Journal of Theoretical and Applied Information Technology
30th November 2025. Vol.103. No.22

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DSS QUERY OPTIMIZATION USING ENTROPYBASED RESTRICTED GENETIC APPROACH ANDPARALLEL PROCESSING

K. SWARUPA RANI[1],B. LAKSHMI[2],Dr. YARLAGADDA ANURADHA[3], Dr. KOLLURU SURESH BABU[4],P. VENKATESWARLU REDDY[5],CH. SABITHA[6], DIVVELA.SRNIVASA RAO[7]

[1]Department of ITPVP Siddhartha Institute of Technology, Andhra Pradesh, India
[2]Department of Computer Applications, School of Science, Siddhartha Academy of Higher Education, Deemed to be University Vijayawada,Andhra Pradesh, India
[3]Department of CSE, Gayatri Vidya Parishad College of Engineering (A) Kommadi, Visakhapatnam, Andhra Pradesh, India
[4]Department of CSE-AIML, Vasireddy Venkatadri International Technological University, Andhra Pradesh, India
[5]Department of CSE, Mohan Babu University (Erstwhile Sree Vidyanikethan Engineering College(Autonomous), Tirupati, Andhra Pradesh, India
[6]Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India
[7] Department of AI & DS, Lakireddy Bali Reddy College of Engineering, Mylavaram, Andhra Pradesh, India.
E-mail: [1]swarupapvpsit@gmail.com,,[2]itslakshmi.h@gmail.com, [3]anuradhayarlagaddag@gvpce.ac.in, [4]kollurusuresh@gmail.com [5]venkateswarlucse@gmail.com, [6]sabithakiran.ch@kluniversity.in [7]srinivassowjanya2012@gmail.com,

ABSTRACT

Any distributed database system's query optimization challenge is exciting and continues to attract a lot of interest. Recent years have seen the application of a number of heuristics that suggest novel strategies for enhancing a query's viability. One of the most widely used heuristics for optimization issues is the Genetic Algorithm (GA). The search for a better solution is still ongoing. This work proposed the Entropy-based Restricted Genetic Approach (ERGA), a novel query optimizer approach. Massive amounts of data (in gigabytes, petabytes, or even more) are processed using DSS queries. As a result, these queries are unlikely to be mission impart antin terms of "Response Time." However, a major concern is the amount of system resources required to execute the query. As a result, Total Costs—the sum of input, output, processing, and communication costs—are used to optimize DSS enquiries. The overall cost of DSS queries can be decreased by increasing the replication factor in the distributed database. It was determined that ERGA more effectively met the low time complexity (Runtime) and high quality (Total Costs) objectives for distributed DSS queries, which are otherwise at odds with one another. Intra-Site Parallelism is proposed to reduce ERGA's performance. Two or more processors may be present at a site in intra-site parallelism.The utilization of two cores or processors over a site has been shown to significantly enhance ERGA quality by up to 6.5%. Additionally, several of the standard metrics of descriptive statistics were used to statistically analyze ERGA. Using ERGA, the best solution outperformed the worst by 89%.

Keywords: *Query Optimization, DSS, Genetic Algorithm, ERGA, Parallel Processing*

1. INTRODUCTION

One new problem in information technology is parallel processing. Several "Jobs" (jobs or Processes) and the computers (Sites or Workstations) that must do these jobs make up a parallel processing environment. These days, multi-core desktop computers are a popular platform for creating intricate, large-scale data-intensive applications.

The advancements in microprocessor technology make them highly suitable for use in computer networks and as a replacement for mainframes and supercomputers, which are exceedingly expensive. A complicated or computationally demanding work is divided into several smaller tasks in parallel processing. To expedite the task's execution, the subtasks are subsequently distributed among several machines for concurrent execution. One of the main research issues in parallel processing is job

allocation or task scheduling.Assigning tasks to the various workstations or locations within a distributed or parallel system is the main goal of task scheduling. Two crucial steps in task scheduling are decomposition and mapping. A complex task is broken down into several smaller tasks in the decomposition process. In a distributed system, mapping is used to assign different subtasks to different locations. Generally speaking, parallel processing makes it easier and more efficient to handle complicated issues. Because multiple processors are used to complete a task, parallel processing speeds up a program.

2. REVIEW OF LITERATURE

Literature lacks discussion on balancing runtime and total costs in DSS query optimizationthis literature review examined a large body of work in the fields of genetic algorithms and query optimization. One of the most prevalent study issues was query optimization, which continues to garner considerable interest. Below is a brief overview of the various important researchers working on query optimization:Tiwari Preeti and Chande Swati V. (2013) examined the various query optimization techniques. The authors recommended using evolutionary approaches as an optimization tool for large and complex queries. Majid Khan, M.N.A. Khan (2013) investigated relational database query optimization strategies. The authors provided a summary of the most important studies conducted by different scholars in the query optimization field. The author presented the advantages and disadvantages of the studies conducted by various researchers.

Subhi H. Hamdoon, Virendra Gawande, and associates (2013) investigated the foundations of query optimization. The authors claim that a query should be optimized if it takes more than a second. The authors list several strategies to enhance queries, including normalizing the database, filtering queries, adding or modifying indexes, substituting stored procedures for queries, modifying database settings, and using query analyzers.In 2013, G.R. Bamnote and S.S. Agrawal illustrated the idea of query processing and how to optimize it. Different Selection and join algorithms were described by the authors. It was also shown how various index types can shorten the time it takes for selection and join operations to execute.

A summary of the basics and characteristics of distributed database systems was provided by Lin Zhou, Yan Chen, et al. (2012). The method of query optimization in distributed database systems was also examined by the authors. Analytically, the impact of using the semi-join operation on a query's optimization process was also noted.

Indu Kashyap, Jyoti Mor, and colleagues (2012) examined query optimization strategies. The authors described Tableaus and Query Graph, two key methods for representing and optimizing a query. The authors found that both transformation and optimization strategies are required to optimize a query containing an aggregate operator (e.g., Sum, Avg, Count, etc.). The study found that it is more difficult to optimize a multi-block query than a single-block query.

A Novel Genetic Algorithm (NGA) was proposed by Ahmat Cosar and Sevinc (2011) to optimize queries in distributed database systems. A collection of OLTP queries in a distributed database was assessed by the authors. The database was duplicated on many accessible locations without employing the fragmentation notion. The impact of different numbers of sites and relations on the optimization performance of a distributed query was noted by the authors. While optimizing the queries, the authors assumed that the sub-operations of a query would be processed sequentially.

Several traditional static and dynamic query optimization techniques were studied by Vijay Raisinghani and Pankit Doshi (2011). The authors found that dynamic programming and iterative dynamic programming are not suitable for query optimization in autonomous distributed database systems. Nonetheless, dynamic programming is the most effective method for query optimization in non-autonomous distributed database systems.

R. Sivaraj and T. Ravichandran (2011) examined a number of genetic algorithm selection methods. The authors discovered that the methods used to choose individuals for crossover and mutation operations vary between the different selection techniques. If a selection process produces distinct individuals for crossover operations to produce the following generation, it is referred to be the finest.

Sambit Kumar Mishra et al. (2011) assessed the costs of various query execution strategies using a genetic algorithm. The authors clarified that in distributed database systems, parallel processing can enhance query performance. The basic features, design, and aspects of distributed database systems were explained by Swati Gupta, Kuntal Saroha, et al. (2011). The definition and function of replication and fragmentation in distributed

database systems have also been clarified by the authors.

One of the best books on genetic algorithms was authored by David E. Goldberg in 2011. The book provides a wealth of knowledge about the fundamentals of genetic algorithms. An effort has been made by the author to provide a basic grasp of genetic algorithms and how they are used in computer science. To create query execution plans for distributed inquiries, T. V. Vijay Kumar and Shina Panicker (2011) employed genetic algorithms. The authors chose a query plan that used fewer sites in order to lower the processing and transmission costs. The authors also looked at how different crossover and mutation probabilities affected a distributed query's optimization process.

The relevance and meaning of query optimization were described by Manoj Kumar Gupta and Pravin Chandran (2011). The author clearly explained the various query optimization techniques. The author also described the Oracle database's optimization procedure. On several platforms (Oracle 8i, 9i, and 10g), the costs of several parameters, such as CPU Time, Elapsed Time, and Disk Access, were calculated and examined.

John Geraghty and Noraini Mohd Razali (2011) explained the various genetic algorithm selection methods. In order to solve the Traveling Salesman Problem, the authors analyzed the effectiveness of the Tournament Selection, Roulette Wheel Selection, and Rank-Based Roulette Wheel Selection strategies. The authors have demonstrated that Roulette Wheel Selection and Tournament Selection are superior to Rank-Based Selection for minor problems.

Using heuristics, Sourabh Kumar, Gourav Khandelwal, and colleagues (2011) presented a novel technique to cost-based query optimization. Based on the results of the simulation, the author came to the conclusion that the heuristic approach outperforms the conventional query optimization methods. In their study, Li Xiafong, Li Dong, et al. (2010) claimed that the performance of distributed database systems is significantly impacted by query processing and its optimization. The usage of semi-joins, as demonstrated experimentally by the authors, successfully lowers the total utilization of system resources by reducing the size of intermediary relations and fragments.

The fundamentals of genetic algorithms have been concisely presented by Manoj Kumar, Mohammad Hussian, et al. (2010). Genetic algorithms, according to the authors, are essentially search heuristics that are used to approximate solutions for optimization issues. The authors described the various applications and methods of genetic algorithms. In their 2010 study, M. Sinha and S.V. Chande concluded that genetic algorithms are a better option for query optimization than deterministic algorithms. The authors promoted the use of genetic algorithms as a method for query optimization in distributed database systems.

The Join Order issue in distributed database systems was examined by S. Vellev (2009). According to the author, a variety of methods, including deterministic algorithms, hybrid algorithms, randomized algorithms, and genetic algorithms, can be used to solve the Join Order problem. The author came to the conclusion that while deterministic algorithms can readily handle small and straightforward questions, Randomize, Genetic, or a hybrid technique should be employed for larger and more complicated queries.

The book by Ozsu and Valduriez (2009) is excellent; it effectively conveys the basic ideas of distributed database systems. The design, architecture, and operation of a distributed database are explained in the book. In order to optimize large and complex queries, Kayvan Asghari, Ali Safari Mamaghani, et al. (2008) recommended using evolutionary techniques. To find the best query allocation strategy for big and complicated inquiries, the authors combined a hybrid approach combining genetic algorithms and learning automata.

Based on a number of performance metrics, Said Elnaffar, Pat Martin et al. (2008) divide the workload and queries of distributed database systems into two categories: OLTP (Online Transaction Processing) queries and DSS (Decision Support System) queries. Queries Ratio, Pages Read, Hit Ratio, Throughput, Number of Locks, and other characteristics are used to classify the workload and queries. According to Swati V. Chande and Madhvi Sinha (2008), genetic algorithms are primarily utilized to solve large and intricate issues. Nutritional Counselling, Robot Trajectory Generation, Aircraft Design, Stylometry, Acoustics, Task Scheduling, Bandwidth Optimization, Query Optimization, and other real-world issues are all addressed by the authors.To address the present need and compare the performance of the Genetic Algorithm with some other widely used query optimization methods,

Reza Ghemi et al. (2008) proposed a multi-agent-based mechanism.

3. METHODOLOGY

The proposed ERGA framework addresses this gap by combining entropy-based GA with parallel processing.ERGA is used to examine a collection of distributed DSS queries that were created in the previous chapter. To address the Operation Site Allocation issue of distributed DSS inquiries, the ERGA concept was put forth. For a distributed DSS query, ERGA's goal was to produce multiple query execution strategies and then choose the most effective one. Two different parallel processing environments—Inter-Site and Intra-Site—were used to optimize the queries. The sub-operations of a distributed DSS query were assigned to various machines that were available at the Inter-Site level. When two or more sub-operations were assigned to a single machine, they were carried out one after the other in a sequential manner.Sub-operations of a DSS query, on the other hand, were assigned to various cores or processors of the machines that were available in the case of the intra-site level. The number of sub-operations that were carried out simultaneously on a single system depended on the number of cores or processors.

Figure1 explains the concept of an environment for inter-site parallel processing. It is evident from Figure 5.1 that a DSS query is first broken down into sub-operations for selection, projection, and joins. The many accessible locations of a distributed database system are then assigned or mapped to these sub-operations.

Figure1: Inter-Site Parallel Processing

According to ERGA policy, the suboperations are allocated; that is, the projection operation was limited to the sites where the matching selection operation was carried out. Maximum entropy served as the basis for site selection.

Lastly, ERGA-generated query execution techniques were used to carry out a DSS query's sub-operations. Each plan's total cost was calculated. The optimum query execution plan was determined to be the one with the lowest Total Costs value. In this case, the amount of parallelism was largely limited by the quantity of machines that were accessible.

Figure 2 explains the intra-site parallel processing environment concept. In this case, multi-core processors were expected at each location. Similar to the Inter-Site environment, the Intra-Site query optimization process begins with decomposition, which is the division of a distributed DSS query into various sub-operations. The sub-operations of a DSS query were decomposed and then mapped onto different distributed database network devices. The assigned sub-operations can be carried out concurrently, depending on the machine's number of cores or processors.

An entropy-based restricted genetic technique was used in several experiments to optimize a set of distributed DSS queries in an intra-site and inter-site parallel processing environment.

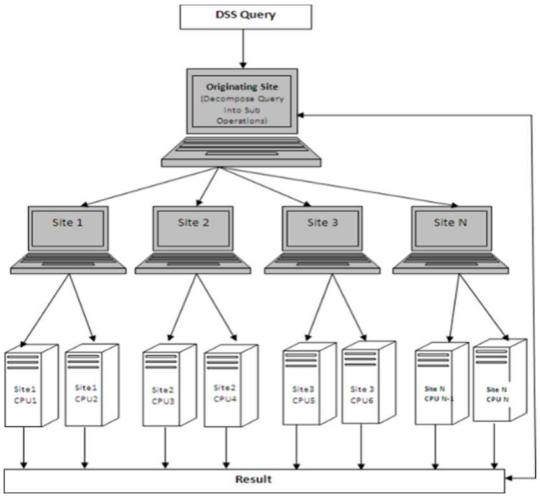

Figure 2: Parallel Processing with Intra Site

In the intra-site parallel processing environment, it is believed that each site has two processors or cores. Two of the sub-operations assigned to a site would be performed simultaneously in order to reduce the overall expenses or execution time per machine. In this instance, the amount of simultaneous processing was limited to

$\sum MiCi.$ where
Mi is a location or the machine.

Ci stands for the machine's number of core processors.

Only when all of a machine's processors are utilized can parallel processing be effective. As a result, "Intra-Site" parallelism is only useful when a

Journal of Theoretical and Applied Information Technology
30th November 2025. Vol.103. No.22

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

distributed database system has at least two or more operations assigned to a single site. If not, it won't significantly lower the DSS query's total costs.

3.1Parameters of Genetic Algorithm

The goal of genetic algorithms is to replicate the fundamentals of biological evolution. The following factors have a significant impact on how a genetic algorithm is designed and operates.1. The size of the population2. The quantity of generations 3.Probabilityof Crossover 4. Probability of Mutations. It is extremely challenging to choose these factors in an optimal combination. It is commonly believed that a legitimate genetic parameter selection may rapidly converge to the best outcomes. However, an incorrect choice can result in a subpar solution or fail to deliver a decent answer.

One crucial genetic algorithm component is population size. It fixes a generation's chromosomal count. It needs to be carefully chosen. There isn't a strict guideline for determining the population size, though. It is often chosen empirically through a series of tests with varying population sizes. When choosing a population size, it is important to remember that a small population does not fully explore the advantages of crossover and mutation operations. However, an excessive population will cause the Genetic Algorithm to perform worse.

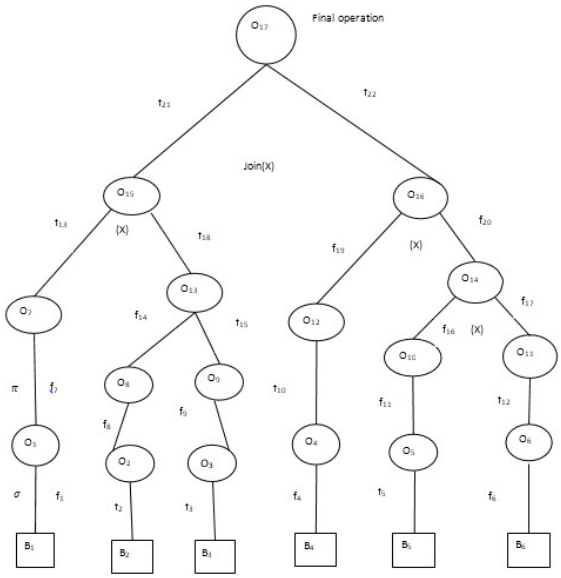

Figure 3: Five Join DSS Query

To examine the impact of different population sizes on the DSS query's total costs, a "Five Join DSS" query was optimized. The range of the population size was 10 to 150. The matching query tree for the 5-Join DSS query is shown in Figure 3.

4. RESULTS AND DISCUSSION

4.1 Impact of Intra-Site and Inter-Site Parallel Processing on DSS Query Total Costs

Table 1 shows the Total Cost values of the various distributed DSS queries obtained using the two different kinds of parallel processing environments**.**Table 1 demonstrates that the use of intra-site parallel processing reduced the overall expenses of a set of distributed DSS queries when compared to an environment that uses inter-site parallel processing. Figure 4 displays the overall expenses for a set of distributed DSS queries when research was done with both intra-site and inter-site parallel processing environments.

Furthermore, it was demonstrated that the use of Intra-Site parallel processing reduced the Total Costs of a set of distributed DSS queries by as much as 6.5% in comparison to an Inter-Site parallel processing environment. In Figure 5.4, this is shown. The total cost of decision support system enquiries decreased as the number of join operations rose.

Table 1: Total Costs with Inter-site and Intra-site Parallel Environment

S. No.	NumberofJoins in a DSS Query	TotalCostswithInter-Site Parallelism	TotalCostswithIntra-Site Parallelism
1.	1	509510	509510
2.	2	1170765	1165824
3.	3	1661635	1639728
4.	4	2172080	2141960
5.	5	2487031	2421643
6.	6	3024110	2875980
7.	7	3393015	3210842
8.	8	4199310	3941458
9.	9	4606150	4304873
10.	10	5222314	4879876

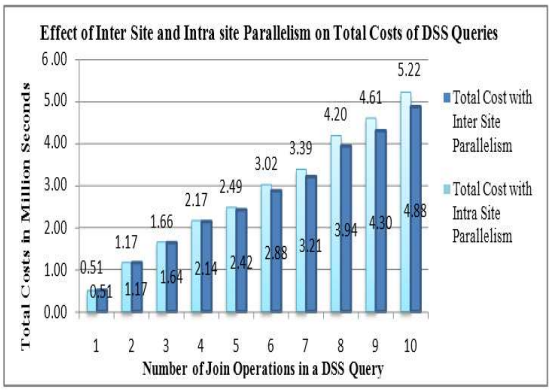

Figure 4: Total Costs of DSS query using Inter & Intra-site Parallel Processing

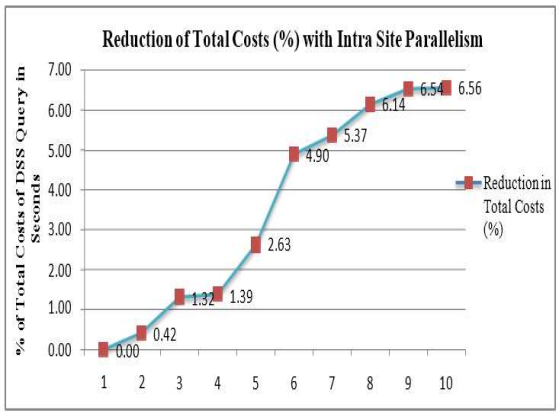

Figure 5: Reduction in Total Costs of DSS query using Intra-site Parallel Processing

8	25	80	2446465	99.01
9	25	90	2446465	99.01
10	25	100	2446465	99.01
11	25	120	2446465	99.01
12	25	150	2446465	99.01

Figure 6, shows how different population sizes affect a five-join distributed DSS query's total costs. According to the experimental findings, the total costs of the distributed DSS query decrease as the population size increases from 10 to 60. Following that, there was no change in the total costs. Thus, 60 is the ideal population size for a Five Join DSS query.

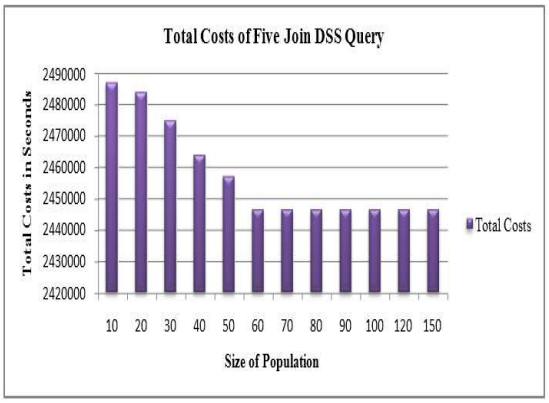

Figure 6 : Total Costs of DSS Query Versus Size of Population

Figure 7, demonstrates how different population sizes affect the quality of the Entropy-based Restricted Genetic Approach solution. According to the experimental results, changing the population size from 10 to 150 resulted in an improvement in the quality of the ERGA solution of about 2%. When the population size exceeds 60, the quality of the solution does not improve; instead, it takes longer to arrive at the best answer.

4.2 Effect of Population Size over Total Costs of DSS Query

Table 2: Total Costs versus Size of Population

S. No.	Number of Generations	Size of Population	TotalCosts	Quality ofSolution
1	25	10	2486715	97.41
2	25	20	2483749	98.86
3	25	30	2474746	97.13
4	25	40	2463746	97.56
5	25	50	2456915	99.03
6	25	60	2446465	99.01
7	25	70	2446465	99.01

Journal of Theoretical and Applied Information Technology
30th November 2025. Vol.103. No.22

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

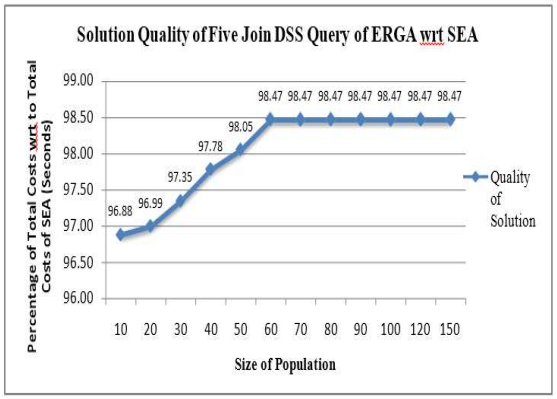

Figure 7: Quality of Solution using ERGA with Varying Size of Population

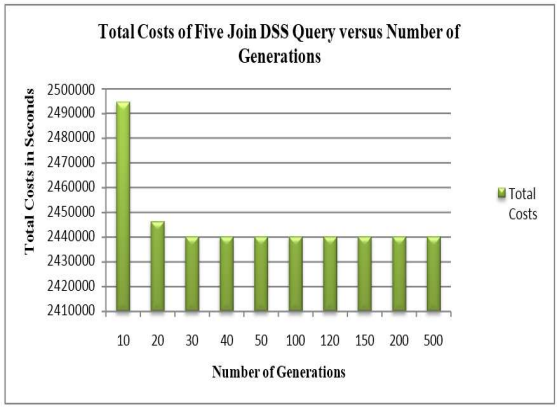

Figure 8: Total Costs versus Number of Generations

4.3 Effect of Number of Generations over Total Costs of DSS Query

In genetic algorithms, the number of generations is crucial. Here, the impact of different generations on the distributed DSS query's total costs is seen. To see how it affected the Total Costs of a five-join DSS query, the population size was set at 60, and the number of generations was changed from 10 to 500.

Table3:Numberof GenerationsVersusSolutionQualityofERGA

S. No.	Size of Population	Number of Generations	TotalCosts	QualityofSolution
1	60	10	2494678	96.57
2	60	20	2446465	98.47
3	60	30	2440346	98.72
4	60	40	2440346	98.72
5	60	50	2440346	98.72
6	60	100	2440346	98.72
7	60	120	2440346	98.72
8	60	150	2440346	98.72
9	60	200	2440346	98.72
10	60	500	2440346	98.72

Figure 8 illustrates how changing the number of generations would affect the Total Costs of a five-join DSS query if the population size were set at 60. Figure 8 shows that the Total Costs of a five-join DSS query decreased up to the 30th generation. When the number of generations was raised from 30 to 500, however, no more change was seen.

From Figure 9, It is deduced that when the number of generations is fixed at 30, the ideal Total Costs are achieved, even though the number of generations can range from 10 to 500. When the number of generations is increased beyond 30, the total costs of the distributed DSS query do not change. Instead, it required additional execution time to deliver the answer.

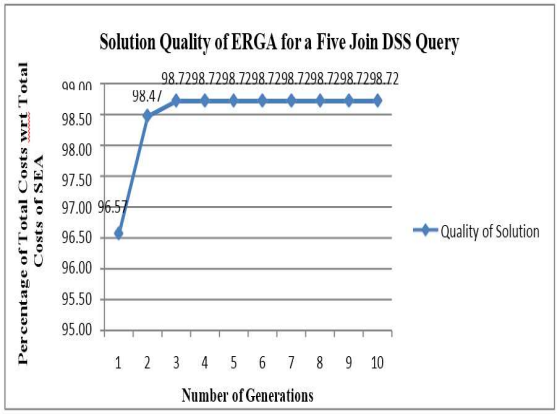

Figure 9: Quality of Solution using ERGA with Varying Number of Generations

4.4 Effect of Crossover and Mutation Probability over Total Costs of DSS Query

Another crucial Genetic Algorithm operation is crossover, which combines two individual chromosomes to produce a new offspring. The crossover operation is followed by a mutation operation. One of the key metrics of the crossover operation is the crossover probability, also known as the crossover rate, which establishes the frequency of the crossover operation, or how many times it was performed. Generally, the crossover operator's probability is represented as Pc, and the crossover probability values fall between 0 and 1. It indicates the percentage of the chromosome pair to

Journal of Theoretical and Applied Information Technology
30th November 2025. Vol.103. No.22

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

be crossed, and if the crossover rate is zero, the offspring will be a perfect replica of their parents. Generally speaking, the crossover rate shouldn't be 100% or zero. While a 100% crossover rate will produce subpar answers, a lesser crossover rate will require longer generations to produce an optimal solution [http://www.obitko.com].

The impact of altering the crossover probability on the total costs of a distributed DSS query was examined through a series of tests. In this part, an Entropy-based Restricted Genetic Approach was used to optimize a five-join DSS query. The crossover rate was varied from 10% to 70% to conduct the experiments. The crossover probability, total costs, and ERGA solution quality values for a five-join DSS query are shown in Table 4.

Table 4: Total Costs versus Crossover Probability

S. No.	Crossover Probability (%)	TotalCostsof DSS Query	Qualityof Solution
1	10	2771070	87.41
2	20	2745915	88.21
3	30	2498952	96.93
4	40	2493260	97.15
5	50	2548760	95.03
6	60	2748760	88.12
7	70	2767915	87.51

From the figure 10, one finding is that, when compared to alternative crossover probabilities, a 40% crossover rate yields more ideal outcomes.

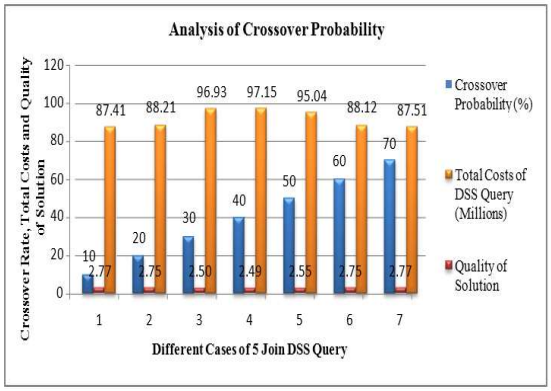

Figure 10: Analysis of Crossover Probability using ERGA

The frequency of the mutation operation is determined by the mutation probability or rate. Additionally, it determines the number of chromosomes that are modified following the crossover procedure. The children are produced immediately following the crossover procedure if the rate of mutation is zero. To improve the characteristics and appearance of the progeny, mutation operations are carried out. The structure of the chromosome produced during the crossover operation will be entirely altered by a 100% mutation rate. Additionally, the GA avoids trapping within a local maximum by using the appropriate mutation rate.Furthermore, a high rate of mutation harms how well a genetic algorithm functions. Because the GA tends to a simple random search when the mutation chance is larger. Typically, fewer than 1% of a generation's population is affected.

Table 5: Total Costs versus Mutation Probability

S. No.	Mutation Probability (%)	TotalCost sof DSS Query	NumberofGeneratio nsto Providean Optimal Solution
1	.5	2771070	50
2	1	2745915	40
3	1.5	2498952	40
4	2	2493260	25
5	2.5	2548760	25
6	4	2748760	40
7	5	2767915	42
8	10	2767915	50

From Figure 11, 2% mutation rate was shown to be the most appropriate for rapidly obtaining an ideal outcome for the DSS query optimization problem utilizing the Entropy-based Restricted Genetic Approach.

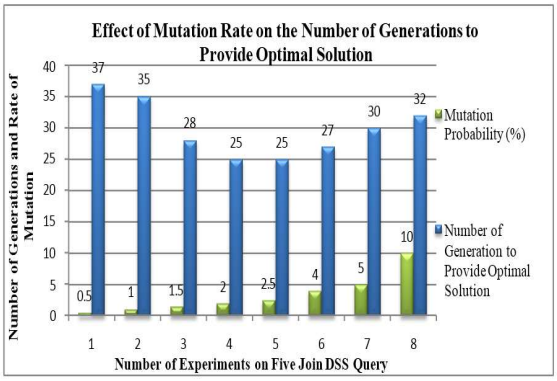

Figure 11: Effect of Mutation Probability over Number of Generations

5. CONCLUSIONS

The study should explicitly state how ERGA's outcomes compare to prior GA-based and heuristic techniques, to emphasize novelty and performance gain

The improvement of ERGA's functionality and design has been the main emphasis of this article. The idea of intra-site and inter-site parallel processing was presented in order to improve the outcomes of ERGA. The degree of parallelism and the quality of the ERGA solution were found to be enhanced by the use of Inter-Site parallel processing. Additionally, a number of genetic characteristics were changed to see how they affected the DSS query's total costs, including population size, number of generations, crossover probability, and mutation probability. It was noted that determining the ideal mix of these characteristics is extremely challenging. There isn't a strict guideline for choosing these settings. Additionally, it has been shown that optimal outcomes can be obtained fast with a well-chosen population size. But a bad choice could result in a subpar outcome. Additionally, there is a cutoff value for both population size and generation count, and no change in the query's total costs was observed above that barrier. An efficient crossover and mutation rate could enhance ERGA's functionality and design even more.

REFERENCES:

[1] Ahmad, I.,Kwok.Yu-Kwong, S Kai-So. "Evolutionary Algorithms for Allocating Data in Distributed Database Systems". Distributed and Parallel Databases, Kluwer Academic Publishers Netherlands. Vol 11. Janary 2002. pp 5-32

[2] Alom B.M. Monjurnul, Henskens Frans, HennefordMicheal."Query Processing and Optimization in Distributed Database System". International Journal of Computer Science & Network Security Vol 9.9 September 2009. pp143-152.

[3] Azari ideh.,"Efficient Execution of Query in Distributed Database Systems". Proceedings of 3rd International Conference on Advanced ComputerTheory and Engineering.Vol 4. August 2010 .pp. 428-433.

[4] Berg C.A.Van Den, Kersten M.L."Analysis of a Dynamic Query Optimization Techniques for Multi-join Queries". Elsevier Journal of System Software. Vol 27. December 1994 . pp. 233-241.

[5] Chen Yan , Zhou Lin, Li Taoying, Yu Yinging. "The Semi-Join Query Optimization in Distriuted Database System". National Conference on Information Technology and Computer Science. Atlantis Press. November 2012. pp. 606-609.

[6] Cornell Douglas W., Philip S.Yu. "On Optimal Site Assignment for Relations in the Distributed Database Environment". IEEE Transactions on Software Engineering. Vol 15.8.Aug 1989 .pp.1004-1009.

[7] Doshi Pankit, Raisinghani Vijay."Review of Dynamic Query Optimization Strategies in Distributed Database". International Conference on Electronics Computer Technology. KanyaKumari.Vol 6. April 2011. pp.145-149.

[8] Elnaffar Said S.A Methodology for Auto-Recognizing DBMS Workload. Conference of the Centrefor Advance Studies on Collaborative Research. Toronto. Oct 2002 pp 1-15.

[9] Gangwani Vinod S., Ramteke P.L. "Query Optimization: Finding the Optimal Execution Strategy". International Journal of Advance Research in Computer Engineering and Technology. Vol 2.2. Feb 2013 . pp . 530-533.

[10] Gupta GK. Database Management Systems. Tata McGraw Hill Education Private Limited. Delhi. April 2011.Chap. 13.

[11] Gupta Manoj Kumar, Chandra Pravin."An Empirical Evaluation of LIKE Operator in Oracle". BIJIT-BVICAM"s International Journal of Computer Applications and Management. Vol 3.2. Dec 2011 pp.351-357.

[12] Harsora Vinay, Shah Apoorva."A Modified Genetic Algorithm for Process Scheduling in Distributed System". International Journal of Computer Applications Special Issue on "Artificial Intelligence Techniques-Novel Approaches&PracticalApplications".2011 special Issue pp .36-40.

[13] Haupt Randy L., Haupt Sue Ellen.Practical Genetic Algorithms. 2nd Edition. Wiley Publishers July.2004. BookChap.1.

[14] Inmon W.H."Building the Data Warehouse". Wiley Publication. University of Michigan. Oct 2005. Chap.1.

[15] J.N. Kapur. Maximum Entropy Models in Science and Engineering. Wiley Eastern Limited. New Delhi. Jan 1989Chap.1.

[16] Jhuno Shim, Peter Scheurmaan and Radek Vingralek."Dynamic Caching of Query Results for Decision Support System". Proceedings of the SSDBM. July 1999 .pp. 1-10.

[17] Kossman Donald"The State of the Art in Distributed Query Processing". ACM Computing Surveys, Vol 32.4. Dec 2000. pp.422–469.

[18] Li Xiafong, Li Dong, Zhi Gao Hong, Yao Lu. "Study of Query of Distributed Database Based

on Relation Semi Joins". IEEE International Conference on Computer Design Application. Qinhuangdao Vol.1. June 2010 .pp.134-137.

[19] Lin Xue. 2009. Query Optimization Strategies and Implementation Based on Distributed Database. Proceedings of the IEEE Transactions on Computer Science, 978-(1)4244-4520. China. Sep 2009 pp.480-484.

[20] Mahajan Sunita, Jadhav P. Vaishali."General Framework for Optimization of Distributed Queries. International Journal of Database Management Systems. Vol 4.3 June .2012 pp. 35-47.

[21] March.S.T, Rho."Allocating Data and Operations to Nodes in distributed Database Design". IEEE Transactions on Knowledge and Data Engineering. Vol 7.2.April 1995 .pp. 305-317

[22] Martin T.P., Lam K.H., Russel Judy I. 1990. "An Evaluation of Site Selection Algorithm For Distributed Query Processing". The Computer Journal.Vol 33.1 Feb.1990 .pp. 61 - 70.

[23] Morrissey J.M., Bandyopadhay S. "Computer Communication Technology and its effects on Distributed Query Optimization". Canadian Conference on Electrical and Computer Engineering. Canada.Sep 1995 pp 596-601.

[24] Nirmeen A. Bahnasawy, Magdy A. Koutb, Mervat Mosa, Fatma Omara."A New Algorithm for Static Task Scheduling for Heterogeneous Distributed Computing System". African Journal of Mathematics and Computer Science Research. Vol 4.6 June.2011 pp. 221-234.

[25] Ozsu M. Tamer, Valduries Patrick. Principles of Distributed Database System. Second Edition, Pearson Education Jan !996. Chap. 1-6.

[26] Omara Fatma A., Arafa Mona M.. "Genetic Algorithm for Task Scheduling Problem". Journal of Parallel and Distributed Computing. Vol 70.1.Jan 2010 pp. 13 -22.

[27] Paulinas Mantas, UšinskasAndrius."A Survey of Genetic Algorithms Applications for Image Enhancement And Segmentation". Information Technology and Control. Vol 36.3. Sep 2007 .pp. 278-284.

[28] Rho Sangkyu, March Salvotre T."Optimizing Distributed Join Queries: A Genetic Algorithm Approach". Annals of Operations Research. Vol 71. Jan 1997 pp.199- 228.

[29] Sharma Charu, Agrawal Pankaj, Gupta Preeti. 2014. Multiple Sequence Alignment with Parallel Computing. International Conference on Advances in Computer Engineering & Applications (ICACEA-2014). Ghaizabad. Feb 2015 pp.16-21

[30] Singh Jasbir, Singh Gurvinder."Task Scheduling using Performance Effective Genetic Algorithm for Parallel Heterogeneous System". International Journal of Computer Science and Tele communication.Vol 3.3. March 2012 .pp.21-26.

[31] Sinha M., Chande S.V. 2010. "Query Optimization using Genetic Algorithms". Research Journal of Information Technology. Vol 2.3 . March 2010. pp. 139-144.

[32] Sivaraj R., Ravichandaran T. 2011. "A Review of Selection Methods in Genetic Algorithms". International Journal of Engineering Sciences & Technology. Vol 3.5.Oct 2011 .pp. 3793 - 3797.

[33] Tiwari Preeti, Chande Swati V. 2013. "Query Optimization Strategies in Distributed Database". International Journal of Advances in Engineering Sciences .Vol 3.3.July 2013 .pp. 23-29.

[34] TPS-DS Benchmark Report. 2012. from website:www.tpc.org/tpcds/spec/tpcds_1.1.0.pdf. accessed on 25/04/2013.

[35] Upadhaya Suchita, Lata Suman."Task Allocation in Distributed Computing VS Distributed Database System: A Comparative Study. International Journal of Computer Science and Networking Security. Vol 8.3 .March 2008 .pp. 338-346.

[36] Vellev S.Review of Algorithm for the join ordering problems in database Query Optimization. Information Technologies and Control.Vol 1 .March 2009 .pp 32- 40.

[37] Ying Peng, Bin Gong_, Hui Liu, and Yanxin Zhang."Parallel Computing for Option Pricing Based on the Backward Stochastic Differential Equation". LNCS 5938, Feb 2010 pp. 325–330.

[38] Zomaya Y Albert, The Yee-Hwei."Observation on using Genetic Algorithms for Dynmaic Load Balancing". IEEE Transaction on Parallel & Distributed Systems. Vol 12.9. Sep 2001 .pp.899-911.