Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/090544

Analysis of DSS Queries using Entropy based Restricted Genetic Algorithm

Manik Sharma[1,*], *Gurvinder Singh*[2], *Rajinder Singh*[2] *and Gurdev Singh*[3]

[1] Punjab Technical University, Kapurthala, India.
[2] DCSE, GNDU, Amritsar, Punjab, India.
[3] Gurkul Vidyapeeth, Banur, PTU, India.

Received: 15 Sep. 2014, Revised: 10 Nov. 2014, Accepted: 5 Dec. 2014
Published online: 1 Sep. 2015

Abstract: Optimization is one of the dominant research areas in the field of different subjects *viz. Mathematics, Computer Science, Business and Economics*.In this paper, an effort has been made to optimize a *Decision Support System (DSS)* query by using the concept of *Exhaustive Enumeration Approach, Dynamic Programming, Genetic Algorithm* and *Entropy based Genetic Algorithm*.The results of different query optimization approaches viz. *Exhaustive Enumeration* (EA), *Dynamic Programming* (DP), *Restricted Exhaustive Enumeration (REA), Simple Genetic Approach (SGA), Entropy Based Restricted Genetic Approach (ERGA)* and *(HC-ERGA) Havrda-Charvat Entropy Based Restricted Genetic Approach* are compared with each other on the basis of *Total Costs, Runtime* and *Quality of Solution*.The concept of *Havrda-Charvat* entropy is used to resolve the low diversity population problem occurs in *Genetic Approach*. The experimental results reveal that when the problem is scaled up *EA*, *DP* and *REA* is intractable to provide an optimal solution for *DSS* queries. Independent of the size and complexity of a *DSS* query, use of entropy with stochastic approach *(HC-ERGA)* provides an optimal solution in a very short and constant time.Furthermore, the results of *HC-ERGA* are more optimal than *EA, DP, SGA* and *ERGA* by 4.7-15.5%, 4.7-15.5%, 6.9-19.5% and 1-4.6% respectively.

Keywords: DSS Query,Total Costs, Entropy, Havrda and Charvat Entropy, Exhaustive Enumeration, Dynamic Programming, Genetic Algorithm.

1 Introduction

The general concept of entropy was proposed by Rudolf Clausius in the year of 1865. In the past few decades, number of definitions and interpretations of entropy has been depicted. Authors from different areas of research defined entropy as degree of freedom, chaos, disorder, measure of uncertainty etc. In information theory, the concept of entropy was given by one of the electrical engineer i.e. Claude Shannon. The objective of his research work was to measure the loss of information while transmitting a message from one end to another. The entropy represents a measure of uncertainty in random variable or random function. Shannon entropy represents the prospect of the logarithms of the probability related with an experiment. For a random variable, entropy represents a function which endeavours to illustrate its unpredictability. It becomes maximum when the distribution of random variable is equal and turned out to be minimum when each random variable has different proportion or unequal probability. Mathematically, entropy of a random variable X with a probability distribution $P(X)$ is maximum when all the probabilities are same and is given as below [1,2,3].

$$H(P) = -k\sum_{i=1}^{n} p_i log p_i \quad (1)$$

Here k is a positive constant. The existing research work reveals that the concept of entropy can be used in different areas like Physics, Biology, Economics, Finance, Sociology, Computer Science (Image Processing, Data Communication, Intelligent Sampling, Cloud Computing, Coding and Compression), Engineering etc. [3,4]

In this research work, an effort has been made to analyze the effect of entropy on a stochastic approach used in the optimization process of a *DSS* query. Here, the stochastic approach is implemented by using the concept

* Corresponding author e-mail: manik_sharma25@yahoo.com

of *Genetic Algorithm (GA)*. *Genetic Algorithm* is one of the evolutionary processes based on the mechanics of natural selection and natural genetics. *GA* is effectively used to solve various computation intensive and optimization problems [5]. The objective of this study is to measure the effect of hybrid approach of *Entropy* and *Genetic Algorithm* in finding an optimal *Operation Site Allocation* plan for a *DSS* query. It is found that the use of fixed level of entropy at *Population/Generation* level of stochastic approach with varying entropy at site selection level significantly improves the optimization process of the DSS query. The use of entropy in stochastic approach reduced the *Runtime* for finding an optimal *Operation Site Allocation* plan of a query to a significant level. The effort is extended to analyze the impact of entropy on different parameters of a *DSS* query by varying some of them for achieving the best possible *Operation Site Allocation plan*.

One of the major limitations of *Genetic Approach* is the local optimal solution. This problem occurs when all of the members (chromosomes) of a generation become similar. This problem creates low diversity population. The concept of *Havrda and Charvat* entropy is introduced to resolve this problem. The objective of using *Havrda and Charvat Entropy* is to reduce the effect of low diversity population so that *Crossover* and *Mutation* operators of *GA* generate a generation with different chromosomes. *Havrda and Charvat* entropy is also known as *Alpha-entropy*.

Mathematically, *Havrda and Charvat Entropy* over a probability distribution P=$P_1, P_2, P_3..P_n$ is defined as

$$H(P) = \frac{1}{1-\alpha} \sum_{k=1}^{n} p^n - 1 \qquad (2)$$

Where Alpha ≥ 0

The remaining part of this section briefly explains the concept of query and its types.

A query is one of the important tools of a database system. In simple terms, it acts as a medium to create, delete and update data in the database. It plays a significant role in the organization and management of the data in the database system. In distributed database systems, there are two major types of queries *viz. Online Transaction Processing (OLTP)* queries and *Decision Support System (DSS)* queries. *DSS* query is one of the important distributed queries, which is normally complex in nature and takes considerable amount of execution time. *DSS* queries consume significant amount of system resources and can saturate even *CPU* or *Memory Server* of the system. One of the major characteristics of *DSS* query is that it's running time is normally unpredictable as compared to *OLTP* queries. In distributed database system, the complexity and distribution of data raison d'etre the need to optimize a *DSS* query. Query optimization is a method of selecting the best *Operation Site Allocation* plan as per optimization function. The distributed queries can be optimized by minimizing either the *Total Costs* or *Response Time* of a query. *Total Costs* are optimized to minimize the usage of system resources or to increase the *Throughput* of the system. On the other hand, *Response Time* is optimized to speed up the execution process of a query [6,7,8,9].

Some of the examples of *DSS* queries are given as below:

- –List names of all the customers who did shopping of average 2.0 lacs in the month of December 2012.
- –Compute the net profit on sales of all the stores situated in Delhi.
- –List the complete city wise information of sales done in the month of December in the state of Punjab and Haryana.
- –Find the complete information of all the customers who spend more money in online shopping rather than on stores.
- –List the details of all the customers who purchased an item with 25% discount.
- –List all the states having at least 20 customers who bought LG LCD in the month of December 2012 with the price rate of at least 25% higher than that it was sold earlier.

2 Related and Proposed Work

Query optimization has been a dominant subject in the field of distributed database system. The past research reveals that the different optimization technique *viz. Exhaustive Enumeration, Dynamic Programming, Branch and Bound, Genetic Algorithm* are used to solve the query optimization problem.

Exhaustive Enumeration approach is one of the deterministic approaches. It performs complete search of solution space through traversal process. It generates and inspects all the possible combinations of search space that is assured to provide the best possible solution. *Exhaustive Enumeration* is simple to understand and implement. However, it is inelegant to solve large and complex problems. In *Exhaustive Enumeration* approach, if one is able to wait long enough, definitely one will get the best solution.

Dynamic Programming (DP) is based on bottom-up approach. It generates the complete sub query execution plan from the sub plans. It is recursive in nature. In general, it uses three different kinds of heuristics with it *viz. Selection Projection, Cartesian Product* and *Tree Form* heuristic. It gives the best solution of the problem after exploring the results of sequence of decisions. For *Operation Site Allocation problem, Dynamic Programming* approach works as below.

- –Characterize the structure of an optimal query execution plan.
- –Recursively identify the value of an optimal query execution plan.
- –Compute the value of an optimal query execution plan using bottom-up approach.

–Design an optimal solution from the calculated information.

Like *Exhaustive Enumeration*, for moderate to complex queries, *Dynamic Programming* is intractable to find an optimal query execution plan in a distributed database system.

Branch and Bound is based upon the concept of search tree. *Depth First Search* is used to find the query execution plan. It is better than *Exhaustive Enumeration* approach. *Branch and Bound* also abbreviated as *B & B* follows recursive approach. In case of an *Operation Site Allocation problem*, it allocates different sites to all possible operations by satisfying the constraint that the current Costs of a query remains less than the current minimum *Total Costs* of a query. Presently, query optimization problem is solved by using different types of evolutionary algorithms *viz. Genetic Algorithm, Ant Colony, Swam Technology* etc[7,10,11,12,13]. In this research work, an effort has been made to analyze a set of *DSS* queries in a novel way by using the concept of *Entropy* and *Genetic Algorithms*. The *DSS* queries are analyzed on the basis of *Total Costs* and *Runtime* of a query. The novel idea of *Entropy based Restricted Genetic Approach (ERGA and HC-ERGA)* is proposed. An effort is made to analyze the effect of entropy for selecting *Size of Population (Hsel(PopSize)* and entropy for selecting site to execute sub operation of a query *(Hsel(Sites))* on the optimization process of *DSS* query. Furthermore, an effort is extended to analyze the effect of *Havrda and Charvat* entropy in resolving the issue of low diversity population problem occurs in the stochastic query optimization approach. The experimental results of the *Havrda Charvat Entropy based Restricted Genetic Algorithm (HC-ERGA)* are compared with the results of *Exhaustive Enumeration (EA) approach, Restricted Exhaustive Enumeration (REA), Simple Genetic Approach (SGA)* and *ERGA* on the basis of *Total Costs* and *Runtime* used in finding an optimal *Operation Site Allocation* plan.

3 Query Processing and Optimization

Query processing is a method that converts a query into an execution plan. One of the foremost objectives of query processing is to convert the query into equivalent set of operations. During the processing, a query goes through several phases like *Scanning, Parsing, Decomposition, Optimization, Code Generation*. One of the major aspects of query processing is to optimize the query. Optimization is one of the mathematical concepts which helps in finding the minima and maxima of functions under certain predefined constraints. One of the major drawbacks of distributed database system is lack of efficiency in handling data access queries. This problem can be resolved by using the concept of query optimization. The core objective of query optimization is to generate numerous query execution plans, and then select one which optimizes either throughput or response time of a query[9,10,13,14].

To optimize a *DSS* query on the basis of usage of system resources, one has to find an optimal query execution plan which minimizes the *Total Costs* of a query. For finding the optimal query execution plan, the costs of different performance metrics *viz. Input-Output Costs, Processing Costs and Communication Costs, Total Costs* and *Runtime* of *DSS* query should be computed and anlayzed. Here, *Total Costs* represents the usage of the system resources required to execute a query. It is also defined as the sum of *Local Processing Costs (LPC)* and *Communication Costs (CMCT)* [15].

Local Processing Costs is the sum of *Input Output Costs* and *Processing Costs of Selection, Projection and Join* operations of query. *Input Output Costs* of a query is one of the dominant parameters of *Total Costs* of a query. It is associated with read and write operations performed between main memory and hard disk of the system. *Processing Costs* of a *DSS* query is associated with the processing of various sub operations of a *DSS* query. *Communication Costs* is one of the important parameters of *Total Costs* of a query in a distributed database. It is defined as the time required to transfer the data from one site to another. Earlier, due to slow communication channels the *Communication Costs* heavily dominated the *Local Processing Costs (Sum of Input Output and Processing Costs)*, hence *I/O Costs* and *CPU Costs* were ignored in the optimization process of distributed queries while computing *Total Costs*[14].

The mathematical formulation of *Local Processing Costs (LPC)* and *Communication Costs (CMCT)* in the distributed *Cost Model* is given as below[12,15]:

$$LPC = \Sigma OA(IOC\Sigma F_y^q M_y^q + CP\Sigma F_y^q M_y^q) \qquad (3)$$

A *DSS* query (q) is decomposed into set of sub operations *(y)* which extracts data from the set of base relations *(b)* which are distributed over a network of sites (s). A query is decomposed into different sub-operations *viz. Selection, Projection* and *Join*.
Here

–OA is Operation Allocation Matrix.
–IOC is Input Output Costs Coefficients.
–CP is Processing Costs Coefficients.
–F is the Intermediate Fragments.
–M is the number of Memory Blocks.

The mathematical representation of the *Communication Costs (CMCT)* is as given below:

$$CMCT_y^q = \Sigma COMM(LPO, JO) * LPF_i + \Sigma COMM(RPO, JO) * RPF_i \qquad (4)$$

Here
COMM :represents the matrix of Communication Costs

from site i to site j.
JO :specifies the location of Join Operation.
LPO and RPO :represents Left Previous Operation and Right Previous Operation.
LPF and RPF :represents Left Previous Fragments and Right Previous Fragment.
NoJ : represent the Number of Join Operations involved in a query. From Eq. (1.3) and (1.4)

$$TCosts(q) = \Sigma OA(IOC\Sigma F_y^q M_y^q + CP\Sigma F_y^q M_y^q) + \Sigma COMM(LPO, JO) * LPF_i + \Sigma COMM(RPO, JO) * RPF_i \quad (5)$$

4 Exhaustive Enumeration and Entropy based Restricted Genetic Approach

In this research work, restricted approach is used with *Exhaustive Enumeration* approach to speed up the *DSS* query optimization process. The approach is restricted as the *Projection* operation are executed on the sites only where the corresponding selection operations is executed. In case of *Exhaustive Enumeration* approach all the different permutations and combinations are explored without using the concept of entropy. *Genetic Algorithm* commonly abbreviated as *GA* is one of the evolutionary algorithms used to solve complex problems. The concept of *GA* was given by *John Holland*. It works on the principle of 'Survival of Fittest'. *Genetic Algorithms* are effectively used for computation intensive and optimization problems. One of the important parameters of *GA* is the fitness function. It is represented as an objective function that may be in the form of a mathematical equation or a subjective function. It can be enumerated as a single-objective function or a multi-objective function. In essence, it defines the objective of the problem that should be optimized. *GA* starts its working with an initial population followed by *Selection, Crossover and Mutation* operators. Selection is an important operator of *GA* that selects the parent from initial population for Crossover to get an effective offspring. *Crossover* operator selects two parents and combines them to get a better offspring. *Mutation* operator further modifies an individual offspring generated by crossover operator [5,16,17]. *Crossover* operator can be implemented by using several techniques like *One-point, Two-point, Shuffle, Matrix* etc. In this paper, One-point Crossover technique is used in which a single site is selected and the bits or characters of parent chromosome next to cross site are swapped. *Mutation* is a unary operator. It alters the selected chromosome. It normally shuffles or alters the bits of characters of the offsprings generated by *Crossover* operator. Technically, it acts as an insurance policy to prevent any type of genetic loss of an individual chromosome (offspring). It is used to avoid the solution to trap into local optima.

In regard to entropy, *Entropy based Genetic Algorithm* with restricted growth encoding scheme (ERGA) is proposed. The novelty of the proposed idea lies in the restricted design of the chromosome. Here, the growth of chromosome *(Operation Site Allocation Plan)* is restricted by using a constraint that the projection operation of query would be performed on the same site where the corresponding selection operation is performed. In case of *ERGA*, the concept of entropy is used at two different levels. Firstly, the concept of entropy is implemented at selection operator of *ERGA*, so that every member of *Population/Generation* has uniform probability of selecting as a parent to perform crossover and mutation operations. The concept of entropy is also used while selecting a site for executing the sub operations of a *DSS* query. Here each permissible site has uniform probability of its selection. Furthermore, in *HC-ERGA*, the concept *Havrda and Charvat Entropy* is used to avoid the low diversity population problem which occurs in the *Genetic Approach*. The low diversity population problem deteriorates the quality of the stochastic approach. The diversity of all chromosome of a population is measured by using the following formula[18].

$$H(P) = \frac{1}{1-\alpha} \sum_{k=1}^{n} p^n - 1 \quad (6)$$

In general, P can be represented as Pij. Here

$$P_{ij} = nS_{ij}/PopSize.$$

nSij represents the number of appearances of site j on the locus of i. *H(P)* approaches to maximum values

$$Max = \frac{PopSize^{1-\alpha} - 1}{1-\alpha}$$

when each sites of a distributed database system involved in DSS query appears uniformly in the population. On the other hand, *H(P)* tends to minimum or zero when all the sites involved in a DSS query lies on the same locus or path of all chromosomes. The pseudo-code of Entropy based Genetic Algorithm with restricted growth encoding scheme is as given below[7,18]:
//**Input Data**

Select the DSS query based upon TPC-DS benchmark database and decompose it into sub queries based upon different operations like selection, projection and join. Read various Input variables viz. NoS (Number of Sites), NoB (Number of Base Relations), NoO (Number of Operations), NoJ (Number of Join Operations), NoF (Number of Intermediate Fragments), NoSo (Number of Selection Operations), NoPo (Number of Projection Operations), IoC (Input Output Costs Coefficients), CP (Processing Costs Coefficients), Comm (Communication Costs Coefficients), PopSize (PopSize),MaxGenr (Number of Generations).

// **Initial Population**

Design chromosome having length one less than the number of operations. Randomly generate an initial

Population by using the concept of Roulette Wheel Selection with PopSize number of Chromosomes.
// **Examine the Diversity of Population**
Call Havrda_Charvat_Entropy
// **Selection Operation with Entropy**
Select any two chromosomes from the initial population having uniform probability that act as parent to perform Crossover and Mutation operations.
// **Crossover Operation**
Apply One-point Crossover operation over two selected parents.
// **Mutation Operation**
Apply Mutation operation on the resultant of crossover operation and store it as a member of new generation.
// **Analyze the fitness**
Total Costs=Input_Output Costs + Processing Costs + Communication Costs
Compute the fitness value of the chromosome based upon Total Costs.
// **Termination**
Generate DSS query allocation plan and Go to step (Analyze the Fitness) until MaxGenr.
//**Procedure to Check the diversity of population using Havrda and Charvat Entropy**
Procedure (Havrda_Charvat_Entropy)
I=0
For J=1 to N

$$H(P) = \frac{1}{1-\alpha}\sum_{n=0}^{k} p^n - 1$$

$$if H(P) \leq = \frac{PopSize^{1-\alpha}-1}{1-\alpha}$$

I=I+1
End if
End for
If I≥ (n/cp)
(Here cp is the control parameter, higher value of cp means better improvement)
Diversity of Population is low
Randomly generate new population based upon Input Parameters by using the Roulette Wheel Selection with average or high diversity
Endif

5 Design of Experiment

A distributed database system based on *TPC-DS (Transaction Processing Performance Council for Decision Support)* benchmark is designed. A simulator is developed in *MATLAB 2008* environment without using the facility of inbuilt *GA* tools. Simulator is able to perform experiments using different query optimization approaches *viz. EA, REA, SGA, ERGA and HC-ERGA*. It is designed on the basis of modular approach. The different modules of simulator are as below

–**Main Program**: Start the simulator and call other modules.
–**Feed Data** : used for feeding the various input parameters.
–**EA-REA** : used for optimizing query using *EA* and *REA*
–**SGA**: used for optimizing query using *Simple Genetic Approach*
–**ERGA and HC-ERGA**: used for optimizing query using *Entropy based Restricted Genetic Algorithm* and by using *Havarda and Charvat Entropy based Restricted Genetic Algorithm*

A set of *DSS* queries is designed on the basis of *TPC-DS* benchmark [7,18]. Each *DSS* query has different *Number of Join Operations*. The design of the various relations and the *DSS* queries is given as below:
Store (Storeid, Sname, Manager, Market, Address, Company, City, State: Varchar; No_of_Emp:Number; S_date: Date;)
Customer (Custid, Cust_Fname, Cust_Lname, DOB, Contact, Email: Varchar;)
Cust_Address (Custid, HouseNo, Street, Street2, City, State, Country: Varchar)
Items (Itemcode, Name, Brand, Type, Size, Colour, Description, Ware_no: Varchar; Price: Number)
Sales(Saleid, Storeid, Store_name, City, Warehouse_id: Varchar; Item_code, Qty, Unit_price, Tax, Discount,Net_price:Number, Custid: Varchar;)
Callcentre (CCID, Cen_name, Manager, Cen_Address: Varchar; No_ofEmp, Area_in_SQFT:Number;)
Webstore (Website, Web_id, Web_mkt_mgr, Nature: Varchar)
Warehouse (Wareh_id, Wname, Wmanager, Address, Company, City, State: Varchar; No_of_Emp, Ware_size: Number; S_date: date;)
Marketing (Mark_id, Mark_item, Mark_promo_name, Mark_Manager, Warehid: Varchar; Expenditure: Number; Mark_sdate, Mark_edate: Date)
Shipping (Ship_id, Ship_mode, Ship_item, ship_address, Ship_cont_person: Varchar; Ship_date: Date; Ship_item_units: Number, itemcode: Number)
Design of DSS Queries
Query 1: DSS 1 Join

$$(\Pi(\sigma)Customer) : X : (\Pi(\sigma)Cust\_Address)$$

Query 2: DSS - 2 Joins

$$(\Pi(\sigma)Customer) : X : (\Pi(\sigma)Cust_Address) : X : (\Pi(\sigma)Sales)$$

Query 3: DSS - 3 Joins

$$(\Pi(\sigma)Customer) : X : (\Pi(\sigma)Sales) : X : (\Pi(\sigma)Warehouse) : X : (\Pi(\sigma)Marketing)$$

Query 4: DSS-4 Joins:

$$(\Pi(\sigma)Cust_Address) : X : (\Pi(\sigma)Sales) : X : (\Pi(\sigma)Sales) : X : (\Pi(\sigma)Cust_Address) : X : (\Pi(\sigma)Sales)$$

Query 5: DSS-5 Joins

$$(\Pi(\sigma)Store) : X : (\Pi(\sigma)Customer) : X : (\Pi(\sigma)Cust\_Addr) : X : (\Pi(\sigma)Store) : X : (\Pi(\sigma)Sales) : X : (\Pi(\sigma)Items)$$

Query 6: DSS-6 Joins

$$(\Pi(\sigma)Sales) : X : (\Pi(\sigma)Cust\_Address) : X : (\Pi(\sigma)Sales) : X : (\Pi(\sigma)Item) : X : (\Pi(\sigma)Marketing) : X : (\Pi(\sigma)Sales) : X : (\Pi(\sigma)Shipping).$$

Query 7: DSS-7 Joins:

$$(\Pi(\sigma)Sales) : X : (\Pi(\sigma)Cust\_Addr) : X : (\Pi(\sigma)Items) : X : (\Pi(\sigma)Warehouse) : X : (\Pi(\sigma)Sales) : X : (\Pi(\sigma)Marketing) : X : (\Pi(\sigma)Shipping) : X : (\Pi(\sigma)Webstore)$$

Query 8: DSS-8 Joins

$$(\Pi(\sigma)Sales) : X : (\Pi(\sigma)Cust_Address) : X : (\Pi(\sigma)Items) : X : (\Pi(\sigma)Warehouse) : X : (\Pi(\sigma)Sales) : X : (\Pi(\sigma)Marketing) : X : (\Pi(\sigma)Shipping) : X : (\Pi(\sigma)Webstore : X : (\Pi(\sigma)Items)$$

Query 9: DSS-9 Joins

$$(\Pi(\sigma)Sales) : X : (\Pi(\sigma)Items) : X : (\Pi(\sigma)Cust\_Address) : X : (\Pi(\sigma)Customer) : X : (\Pi(\sigma)Store) : X : (\Pi(\sigma)Sales) : X : (\Pi(\sigma)Warehouse) : X : (\Pi(\sigma)Sales) : X : (\Pi(\sigma)Marketing) : X : (\Pi(\sigma)Sales)$$

Query 10: DSS-10 Joins

$$(\Pi(\sigma)Sales) : X : (\Pi(\sigma)Items) : X : (\Pi(\sigma)Cust\_Addr) : X : (\Pi(\sigma)Customer) : X : (\Pi(\sigma)Store) : X : (\Pi(\sigma)Sales) : X : (\Pi(\sigma)Warehouse) : X : (\Pi(\sigma)Sales) : X : (\Pi(\sigma)Marketing) : X : (\Pi(\sigma)Sales : X : (\Pi(\sigma)shipping)$$

It is quite difficult to set the values for genetic parameters as the values of these parameters significantly affect the optimization process. For *SGA, ERGA and HC-ERGA* number of experiments are conducted by varying the different parameters of genetic approach like *Size of Population, Number of Generations, Crossover Rate, Mutation Rate* etc. Empirically, it is observed that the optimal value of *Total Costs* for the above set of DSS queries is obtained with the following statistics of genetic parameters.

–Size of Population (PopSize) : 50
–Number of Generation (MaxGenr) : 50
–Crossover Probability : 0.3
–Mutation Probability : 0.02
–Length of Chromosome : Number of Operations-1

6 Effect of Number of Join Operations and Entropy over Total Costs of DSS Query

Join is one of important operations of Relational Algebra. It is used to combine two or more relations to create a new relation. It is costly in terms of usage of system resources in context to a distributed database. An effort has been made to analyze the effect of *Number of Join Operations* over the *Total Costs* of *DSS* query using different query optimization approaches. Number of experiments are performed for a set of *DSS* queries, to compute and analyze the *Total Costs* and *Runtime* of *DSS* queries by using various approaches *viz. EA, DP, SGA, REA, ERGA and HC-ERGA*. In *Genetic Approach (SGA, ERGA and HC-ERGA)* each experiment is conducted ten times and the average of the results is taken in Table 1.Table1 shows the output of the various experiments performed by using different optimization approaches over a set of *DSS* queries having *Number of Join* operations ranging from 1 to 10.

Table 1: Analysis of Total Costs

EA	DP	SGA	REA	ERGA	HC-ERGA
535107	535107	547723.3	500100	509510	509510
1249428	1249428	1258572	1167690	1184765	1184765
1788080	1788080	1786258	1655630	1681635	1664819
2311937	2311937	2367567	2127680	2172080	2150359
2625924	2625924	2673558	2409105	2487031	2462161
3172274	3172274	3250918	2883885	2924110	2865628
3714574	3714574	3784998	3361605	3393015	3325155
4313116	4313116	4390830	3885690	3899310	3782331
5121945	5121945	5304871	4573165	4606150	4467966
6040930	6040930	6194513	5119432	5222314	4982088

From the above tabular statistics and Figure 1(a), it is observed that independent of the query optimization approaches, *Total Costs* of *DSS* query increases with an increase in *Number of Join Operations*. *Total Costs* of query as given by *REA* is always less than as that of *EA, DP, SGA, ERGA and HC-ERGA*. Hence, *Restricted Exhaustive Enumeration* approach always yields to the best *Total Costs* for *Operation Site Allocation* plan as Genetic and Entropy based query optimization approaches.

In context of entropy, as the size of population is fixed for each experimental query, therefore the entropy (Hsel(PopSize)) of different members of population/generation of selecting them as a parent remains constant. On the other hand, the entropy of selecting a site for executing the sub operations of a *DSS* query *(Hsel(Sites))* increases with an increase in *Number of Sites*. Table 2 shows the values of *Entropy Hsel(PopSize)* and Hsel(Sites) for a set of DSS queries as designed earlier. From the experimental results, it is observed that an increase in Entropy[2] Hsel(Sites) reduces the *Total Costs* of *DSS* query by 4.67-13.22% and 6.98-15.69% as compared to *Exhaustive Enumeration (EA)* and *Simple Genetic Approach (SGA)*.

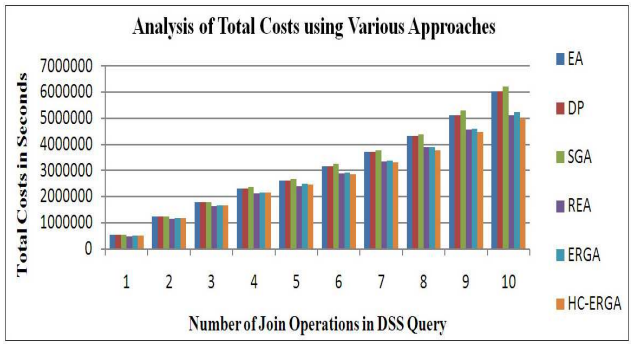

Figure 1(a): Total Costs using different Query Optimization Approaches

Table 2: Analysis of Entropy over Size of Population and Number of Sites

Joins	Sites	Entropy[1]	Entropy[2]	TotalCosts
1	2	84.94	0.60206	509510
2	3	84.94	1.431364	1184765
3	4	84.94	2.40824	1681635
4	6	84.94	4.668908	2172080
5	6	84.94	4.668908	2487031
6	6	84.94	4.668908	2924110
7	8	84.94	7.22472	3393015
8	8	84.94	7.22472	3899310
9	10	84.94	10	4606150
10	12	84.94	12.95017	5222314

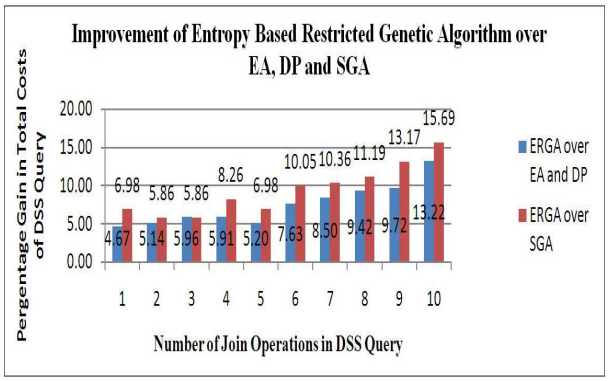

Figure 1(b): Improvment of ERGA

Figure 1(b) shows the improvement of *Entropy based Restricted Genetic Algorithm* over *Exhaustive Enumeration (EA), Dynamic Programming (DP) and Simple Genetic Algorithms (SGA).*

Furthermore, Table 1 shows that the results of *Havrda Charvat Entropy based Restricted Genetic Approach (HC-ERGA)* outperform the results of *EA, DP, SGA and ERGA.* From Figure 1(c), it is clear that the result of *HC-ERGA* are better than *EA, DP, SGA and ERGA* by 4.7-15.5%, 4.7-15.5%, 6.9-19.5% and 1-4.6% respectively.

From the Figure 1(a) and 1(c), it is clear that the use of *Havrda and Charvat entropy with Restricted Genetic Approach* gives more optimal value of *Total Costs* of the *DSS* query as compared to *Exhaustive Enumeration, Dynamic Programming, Simple Genetic Approach and Entropy Based Restricted Genetic Approach.* In other words, the use of *Havrda and Charvat Entropy* resolves the low diversity population problem to a great extent.

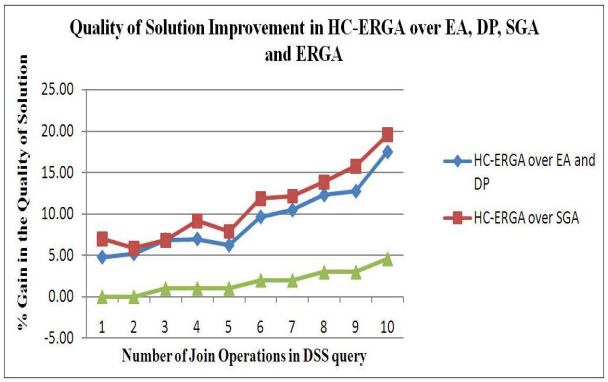

Figure 1(c): Quality of Solution Improvement in HC-ERGA.

6.1 Advantages of Havrda and Charvat Entropy based Restricted Genetic Approach

From the above results in general, it is concluded that

- –*HC-ERGA* is a query optimization approach based upon Entropy and Genetic Algorithm.
- –It provides better optimal results as compared to *Exhaustive Enumeration, Dynamic Programming, Simple Genetic Approach and Entropy based Restricted Genetic Approach.*
- –It avoids the creation of duplicate chromosomes while generating a new generation.
- –It resolves the problem of low diversity population.
- –Results can be obtained very quickly as compared to *Exhaustive Enumeration approach.*

7 Effect of Number of Join Operations and Entropy over Runtime of DSS Query

Time taken *(Runtime)* in providing an optimal *Operation Site Allocation* plan is one of the important parameters of any query optimization technique. Query optimization must be able to provide the optimal solution in small and finite amount of time.

Figure 2 shows the graphical representation of *RunTime* of different query optimization approaches used in finding

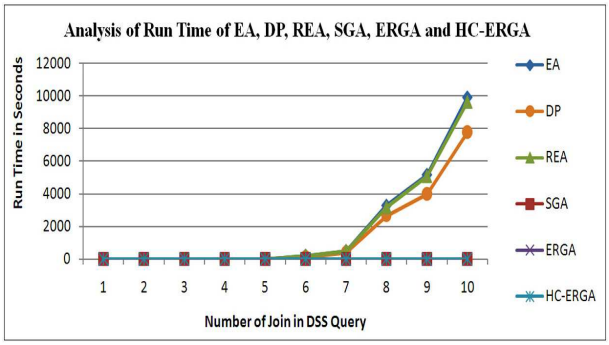

Figure 2: Run Time for finding optimal solution using different optimization techniques

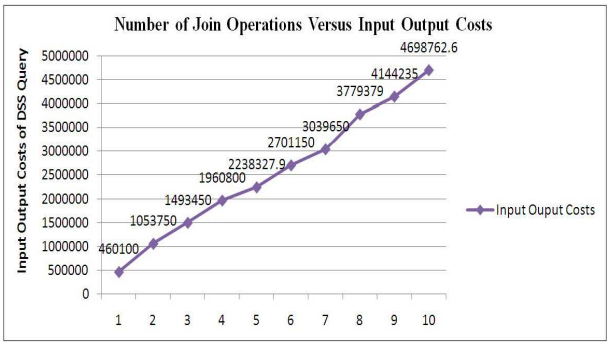

Figure 3: Number of Join Operations Versus Input Output Costs

an *Operation Site Allocation* plan for a set of *DSS* queries. In *EA* and *REA*, it is found that there is an exponential relationship between the *Number of Join Operations* and the time required to provide the optimal solution.

From Figure 2, it is inferred that the *Exhaustive Enumeration, Dynamic Programming and Restricted Exhaustive Enumeration* approach became intractable to find the solution for large and complex *DSS* queries as the run time increases to significant level. *Dynamic Programming* takes lesser time than *Exhaustive Enumeration*, but still shows an exponential behaviour along with the *Number of Join Operations*.

On the other hand, *Havrda and Charvat Entropy based Restricted Genetic Approach (HC-ERGA)* provides a solution very close to *REA* in constant time. For medium to large *DSS* query, gain in execution time (Runtime for providing an optimal solution) in *HC-ERGA* lies in the range of 5-10000 seconds. However, no significant Runtime difference is observed when results of *HC-ERGA* are compared with *SGA* and *ERGA*. From Figure 2, it is clear that the Runtime graph line for *SGA, ERGA and HC-ERGA* are almost overlapped on each other. Therefore, in stochastic and entropy based approach, the time required to provide solution is independent of the *Number of Join Operations*.

8 Effect of Number of Join Operations over I/O and Communication Costs in HC-ERGA

In *Havrda and Charvat Entropy based Genetic Approach* with restricted growth encoding scheme, number of experiments are conducted to analyze the relationship between *Input Output Costs, Communication Costs* with *Total Costs* of a query. In the experimental process, the *Communication Costs* is fixed with respect to the *Input Output Costs* to analyze the set of *DSS* queries. The ratio of *Input Output Costs* to *Communication Costs* was taken as 1:1.6 [24]. Table 3 shows the experimental values of *Input Output Costs, Communication Costs and Total Costs* of different queries as obtained in an optimal *Operation Site Allocation* plan.

Table 3: Communication Costs and Total Costs for a set of DSS Queries using ERGA

S.No	Joins	IO_Costs	Comm_Costs	Total_Costs
1	1	460100	3800	509510
2	2	1053750	12600	1184765
3	3	1493450	20200	1664819
4	4	1960800	24200	2150359
5	5	2238327.9	29871	2462161
6	6	2701150	40600	2865628
7	7	3039650	44600	3325155
8	8	3779379	50400	3782331
9	9	4144235	61890	4467966
10	10	4698762.6	81060	4982088

From Table 3, it is found that *Input Output Costs* plays significant role in the *Total Costs* of a *DSS* query. *Input Output Costs* increases exponentially with an increase in *Number of Join Operations*. The *Input Output Costs* has been raised to ten times when the *Number of Join* operations are raised from 1 to 10. Figure 3 represents the exponential relationship between *Number of Join Operations* and *Input Output Costs* of a *DSS* query. The percentage of *Input Output Costs* in *Total Costs* of a query remains almost constant for each *DSS* query.

Furthermore, from Table 3, it is found that *Communication Costs* increases with an increase in *Number of Join* Operations. There exists an exponential relationship between the *Number of Join* operations and the *Communication Costs*. Figure 4 shows the graphical representation of *Communication Costs* along with *Number of Join Operations*.

From Figure 4, it is found that the percentage of *Communication Costs* in *Total Costs* of query varies from

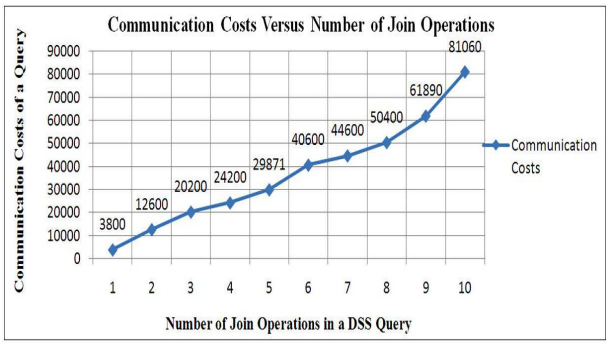

Figure 4: Analysis of Number of Join Operations over Communication Costs

0.75% to 1.4%. Therefore, it is concluded that the percentage of *Communication Costs* in *Total Costs* of a query also increases with an increase in *Number of Join Operations*.

Conclusion

Optimization is one of the vital tools which assists in analyzing the physical and logical systems. In terms of mathematics, an optimization problem is a decisive problem, which finds the best possible solutions from the available set of solution. This paper introduced the process of distributed query optimization in a simple and intuitive way. An innovative idea of restricted approach in *Operation Site Allocation* plan has been depicted. This paper depicts the use of *Entropy based Genetic Algorithm* in the optimization process of the *DSS* query. *Hsel(PopSize)* represent the entropy of Popsize/Generation of Genetic Approach. As the size of population is fixed and same for each experimental query therefore, the entropy for selecting the parent from the population or generation of Genetic Approach remains constant. On the other hand, the *(Hsel(Sites))* represents the entropy of selecting a site for executing the sub operation of a *DSS* query. It increases with the *Number of Sites*. An innovative idea of restricted growth in *Operation Site Allocation* plan has been implemented. *Restricted Exhaustive (REA) and Entropy based Genetic Approach (ERGA and HC-ERGA)* approaches have been used for optimizing the *DSS* queries in a distributed environment. In *HC-ERGA*, the effect of *Number of Join* operations on the different performance metrics namely *Input Output Costs*, *Processing Costs*, *Communication Costs*, *Total Costs* and *Runtime* of the *DSS* query is also examined.

From the experimental results, it is found that *EA* and *REA* explore all the permutation and combination to generate the different *Operation Site Allocation* plans. Hence, the solution space of *REA* rises exponentially when size of the problem is scaled up. The increased size of solution space takes longer time to provide an optimal query execution plan. Therefore, it became intractable to find the solution for large and complex *DSS* queries using *EA* and *REA*. *Dynamic Programming* shows exponential *Runtime* behaviour with *Number of Join Operations*. Therefore, *DP* is also incapable to provide an optimal result for moderate to complex *DSS* queries. On the other hand, in case of *Havrda and Charvat Entropy based Restricted Genetic Approach*, the solution space remains constant as it depends upon the *Size of Population* and the *Number of Generations*. Therefore, *HC-ERGA* provides solution very close to *REA*. One of the interesting factors of using *HC-ERGA* in the optimization process is that one can get the optimal *Operation Site Allocation* plan in very short and almost constant time, independent of the complexity of the *DSS* query. It is also found that *Total Costs* of the system resources increases with the increased *Number of Join* operations. An exponential relationship between the *Number of Join* operations and the *Total Costs* of a *DSS* query is observed. It is found that an increase in $Entropy^2$ *Hsel(Sites)* helps in reducing the *Total Costs* of *DSS* query by 4.67-13.22% and 6.98-15.69% as compared to *Exhaustive Enumeration (EA)* and *Simple Genetic Approach (SGA)*. The use of *Havrda and Charvat Entropy* assists in reducing the low diversity population problem to a significant extent. The results of *HC-ERGA* are better than *EA, SGA and ERGA* by 4.7-15.5%, 6.9-19.5% and 1-4.6% respectively. Therefore, the use of *Havrda and Charvat Entropy* improves the quality of solution of stochastic approach for the optimization of the *DSS* queries.

Input Output Costs of a query is one of the dominant parameters of *Total Costs* of a query. *Input Output Costs* increase exponentially with an increase in *Number of Join Operations*. An exponential growth with ratio of 89.3-90.3% in *Input Output Costs* of the *Total Costs* of the *DSS* query is also revealed. Furthermore, in *Entropy based Restricted Genetic Approach*, an exponential relationship between the *Number of Join Operations* and the *Communication Costs* is also witnessed. In the experimental set of DSS queries, in which the *Number of Join Operations* is varied from 1 to 10, the ratio of *Communication Costs* in *Total Costs* of query varies from 0.75% to 1.4%.

Acknowledgement

Authors are highly thankful to the Department of RIC, Punjab Technical University, Kapurthala, Punjab, India.

Conflict of Interest

Authors declare no conflict of interest.

References

[1] J.N. Kapur. Entropy and Encoding. Mathematical Sciences Trust Society, New Delhi. pp. 16.

[2] Hien To, Kuorong Chiang, Cyrus shahabi. Entropy-based Histogram for Selectivity Estimation. CIKM 2013. pp. 1939-1948.

[3] Rongxi Zhou, Ru Cai, Guanqun Tong. Applications of Entropy in Finance: A Review. Entropy 2013;15:4909-4931.

[4] Eduardo J.Solteiro Pires, Jose A. Tenreiro Machado, B. De Moura Oliveira. Entropy Diversity in Multi Objective Particle Swarm Optimization. Entropy 2013;15:5475-5491.

[5] David E Golberg. Genetic Algorithm in Search, Optimization and Learning. Seventh Impression. Pearson Education. pp. 1-2.

[6] Clark D. French. One Size Fits All- Database Architecture Do Not Work for DSS. ACM SIGMOD Newsletter1995:24-2:449-450.

[7] Manik Sharma, Gurvinder Singh, Rajinder Singh , Gurdev Singh. Stochastic Analysis of DSS Queries for a Distributed Database Design. International Journal of Computer Applications (IJCA) 2013.; 82-5:36-42.

[8] TPC Benchmark DS, Version 1.1.0, April 2002 http://www.tpc.org Accessed on June 2013.

[9] Said Elnaffar, Pat Martin et. Al.. Is it DSS or OLTP: Automatically Identifying DBMS Workload. Journal of Intelligent Information System 2008; 30: 249-271.

[10] Narasimhaiah Gorla, Suk-Kyu Song. Subquery Allocation in Distributed Database using GA. Journal of Computer Science and Technology. 2010; 10-1:31-37.

[11] T.V.Vijay Kumar, Vikram Singh. Distributed Query Processing Plans Generation Using GA. International Journal of Computer Theory and Engineering 2011; 3-1:38-45.

[12] Sangkyu Rho, Salvatore T. March. Optimizing Distributed Join Queries: A GA Approach. Annals of Operation Research 1997; 71:199-228.

[13] Ahmet Cosar, Sevinc Endvic. An Evolutionary Genetic Algorithm for Optimization of Distributed Database Queries. The Computer Journal . 2011; 54: 717-725.

[14] S. Vellev. Review of Algorithms for the Join Ordering Problems in Database Query Optimization. Information Technologies and Control 2009; 1:32-40.

[15] M. Tamer Ozsu and Patrick Valduries. Principles of Distributed Database System. Second Edition 2009, Pearson Education, pp.160.

[16] M. Sinha, SV Chande. Query Optimization using Genetic Algorithm. Research Journal of Information Technology 2010; 2-3:139-144.

[17] Zehai Zhou. Using Heuristics and Genetic Algorithm for Large Scale Database Query Optimization. Journal of Information and Computing Science 2007; 2-4:261-280.

[18] Baljit Singh, Arjan, Akashdeep. Havrda and Charvat Entropy Based Genetic Algorithm for Travelling Salesman Problem. International Journal of Computer Science and Network Security 2008;8.5:312-319.

[19] Yufang Qin, Junzhong Ji, and Chunnian Liu. An Entropy-Based Multi-objective Evolutionary Algorithm with an Enhanced Elite Mechanism. Applied Computational Intelligence and Soft Computing Volume 2012; Article ID 682372; 1-11.

[20] Mohamed A. El-Sayed, Hamida A. M. Sennari. Multi-Threshold Algorithm Based on Havrda and Charvat Entropy for Edge Detection in Satellite Grayscale Images. Journal of Software Engineering and Applications, 2014, 7, 42-52.

[21] Milivoje M. Kostic. The Elusive Nature of Entropy and its Physical Meaning. Entropy 2014;16:953-967.

[22] Robert M. Gray. Entropy and Information Theory. Springer Verlag, USA. pp. 21-37.

[23] J.N. Kapur. Maximum Entropy Models in Science and Engineering. Wiley Eastern Limited, Revised Edition 1993 pp. 1-20.

Manik Sharma is pursuing his doctrate form Punjab Technical University under the kind guidance of Dr. Gurdev Singh, Dr. Gurvinder Singh and Dr. Rajinder Singh. His areas of interst are Distributed Database, Query Optimization, Genetic Algorithm and Metrics.

Gurvinder Singh is working as a Professor and Head, Department of Computer Science and Engineering, GNDU, Amritsar. His areas of specialization are parallel and distributed computing. Author has around 18 years of teaching experience. Author has number of national and international research paper to his credit.

Rajinder Singh is working as a Associate Professor in the Department of Computer Science and Engineering, GNDU, Amritsar. His areas of specialization are distributed database and soft computing. Author has around 17 years of teaching experience.

Gurdev Singh is doctorate in Computer Scoience and Engineering. He has around 8 years or experience in teaching graudeate and post graduate classes. An author has number of research paper to his credits. His areas of interests are Software Metric, Database and Operation Research.