Volume 3, No. 5, May 2012
Journal of Global Research in Computer Science
ISSN-2229-371X
RESEARCH PAPER
Available Online at www.jgrcs.info

# OPTIMIZED GENETIC ALGORITHM FOR SPATIAL DATA EVENTS IN ACTIVE DATA WAREHOUSE

*Paramasivam . K and Dr.C.Chandrasekar
*Research scholar and Associate Professor,
Dept. of Computer Science,
Periyar University, Salem, TamilNadu -India.
param_sivam@yahoo.com

***Abstract:*** A fast changing in dynamic heterogeneous environment makes the data to be maintained up-to-date. Due to enormous amount and composite formation of spatial data, the optimization of spatial query is a complex process and significant from research viewpoints. Numerous programming techniques and methods are being presented to optimize the solutions formed by spatial analysis tools. The previous work used the GA technique for multi-join operation in active data warehouse. Genetic Algorithm (GA) is one of the normal optimization schemes, which creates many possible optimal solutions than the extra linear programming tools. But the downside of the previous work is that it does not discuss about the spatial data transformation and execution. To overcome the issues raised over spatial data in active data warehouse, in this work, we are going to present a new technique termed as optimized genetic algorithm used to form logical entity relation with spatial state of data extraction and transformation. Several query operations are carried over with spatial data to perform multi-join relations in active data warehouse and the selection of more appropriate combination of multiple relations are done by using optimized genetic algorithm. An optimized crossover and optimized mutation chooses the best combination of multiple relations of joins contains spatial data. Experimental evaluations are carried out with both synthetic and real datasets to estimate the performance of the proposed optimized genetic algorithm for multi-join relation with spatial data extraction and transformation in terms of multi-join execution time, optimal query time and gene population are the metrics being used to compute optimal threshold for multi-relational joins generation.

***Key words:*** Spatial data, Active data warehouse, multi-join operation, optimized GA, multi-relational join

## INTRODUCTION

A sequential or active study of spatial data is desired in diverse fields such as ecological structure examination. One of the largest parts of primary problems which users are facing is the complexity in produce spatio-temporal field of eminence data for study during an interpolation or combination of observational data. In several fields, to develop consistency of spatio-temporal exclamation in producing eminence data, models and/or equations recounting an essential method and structure is incorporated with observational data. By combining observational data and methods recounting primary approaches and construction of object-phenomenon with a GIS, we can present a GIS-based situation which permits active update of spatio-temporal field of data at any time a novel observational data and an improvement of representations are given. If computational speed for combination is fast adequate, we can accumulate only observational data and can guess data at any position based on the requests.

Spatial data is a phrase used to illustrate data that supports to the space engaged by substance in databases. This data is algebraic and is diverged. It comprises points, rectangles, lines, polygons, volumes, surfaces, as well as time, and data of yet advanced dimension. Spatial data is regularly established in concurrent with non-spatial data. Spatial data can be distinct or uninterrupted. When it is distinct, then it can be represented using conventional systems from relational DBMSs. Genetic Algorithms are dissimilar from conventional methods in various ways, as they effort with a system of the parameters and not with the parameters themselves. They searched from a inhabitants of points not a particular point. They have been modified for resolving a range of operational research problems. An outline of Genetic Algorithm (GA) is given in Figure1

```
Random generation of Initialize population
Evaluate Fitness of individuals
Repeat
        Rank individuals and allocate reproduction trials
        Until max. no. of population initialized
        Randomly select 2 parents from the list
        Apply crossover and mutation
        End
Evaluate fitness of off-springs
Until (condition satisfied)
Solution = saved best feasible and the fittest
```

Figure 1 Genetic Algorithm Outline

In order to construct Genetic Algorithm well-organized, it is customized to keep the greatest so far solution, this variation is called Modified Genetic Algorithm (MGA). GIS is utilized as a spatial study tool to situate and discover how many regions are appropriate for the given spatial data. An optimization technique has been used for spatial data transformation techniques.

In this paper, an optimized optimization technique is used here for the development of multi-join operation among the spatial data transformation and execution. The spatial data events are monitored and joined by an optimized GA model and described briefly in section 3.

## LITERATURE REVIEW

To keep the data up-to-date with the recent activities in data warehouse, several techniques have been presented earlier to perform the several numeric operations on the active data warehouse. Based on active data warehouses, the variation made in the data using streams is pressented by scalable [8] scheduling of streaming data streams. Active Data Warehousing has happened as an replacement to conventional warehousing, so the constraint for on-line warehouse stimulant institutes frequent confronts in the completion of data warehouse transformations, propose a [9] specialized join algorithm, termed mesh join (MESHJOIN), that gives back for the distinction in the entrée cost of the two join inputs. A Joint Optimization Algorithm [10] is used for redistribution tasks in agent-based workflow organization systems which has been adapted to an automatic adaptive migrations [4].

For performing the numeric operations on data warehouse, several techniques have been used. The most probably used one is Genetic algorithm [5] which has been presented for spatial state of configurations across the network. Using the genetic algorithm, the optimized [7] index tracking system is done efficiently on data warehouse but it consumes more amount of time in order to perform it. Another author presented genetic algorithm for genetic search [6] and the GA features are also been used for heart disease diagnostics [1]. An improved genetic algorithm [2] is also being used to perform the search on the active data warehouse and the set of events which are occurred on the tables are also been accessed with General hybrid column generation algorithm for crew scheduling problems using genetic algorithm [3]. In this work, we are going to implement an optimized GA technique for multi-join operation in active data warehouse. Optimization is done based on the logical multi-join relational entity of the stream tuples present in the table.

## OPTIMIZED GA FOR SPATIAL DATA EVENTS IN ACTIVE DATA WAREHOUSE

The proposed work used Optimized Genetic algorithm which is efficiently designed for the process if identifying the spatial data transformation and execution in a particular interval of time. The proposed optimized GA comprises of two operations. The first operation is to identify the spatial data transformation and execution at a particular interval of time. The next operation is to identify the logical entity relationship of the spatial data using an optimized GA.

The process of identifying the spatial data transformation is done in the active data warehouse. A widespread method to compact with spatial data is to hoard it clearly by parameterize it and thereby attaining a decline to a position in a probably superior dimensional space. The preservation and revolution of spatial data concerns the capability to enter, operate, and change data once it has been produced. Though many diverse analysis subsist with respect to what comprises these abilities can be recognized.

The second process is used to identify the logical entity relationship of multi-join operation with the spatial data. Several query operations are carried over with spatial data for multi-join relations in active data warehouse and the selection of more appropriate combination of multiple relations are done by using optimized genetic algorithm. A crossover and mutation chooses the best combination of multiple relations of joins for identifying the logical entity relationship. The process of the optimized GA for spatial data extraction in active data warehouse is shown in fig 3.1.

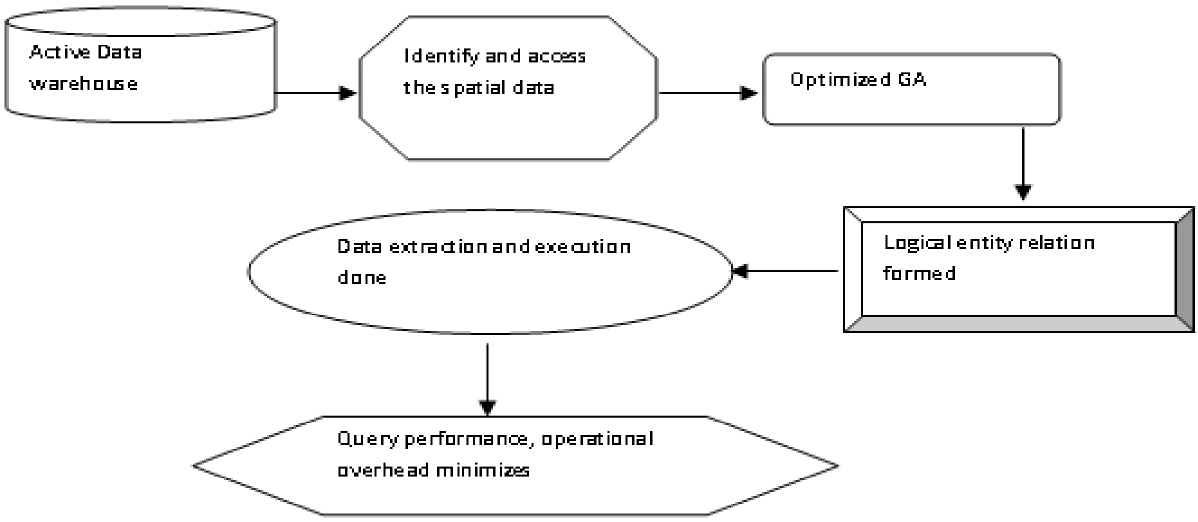


Figure 2 Architecture diagram of the proposed optimized GA for spatial data extraction in active data warehouse

The process of extracting the spatial data are identified through active data warehouse and the logical multi-join entity relationship is done through optimized Genetic algorithm which selects the best multi-join relational entity among the spatial data in the active data warehouse.

### *Optimized Genetic Algorithm for Logical Multi-Join Relational Entity:*

The proposed optimized GA is mainly used to identify the optimization is based on logical entity relation formed with spatial state of data extraction and transformation. The optimized selection is done with an appropriate multi-join relations based on spatial data extraction and execution. As like optimized selection, an optimized crossover is done to provide a new generation of sample multi-join operations in the search space and an optimized mutation is done to ensure the genetic diversity within the population. The process of optimized genetic algorithm is shown in fig 3.2.

 

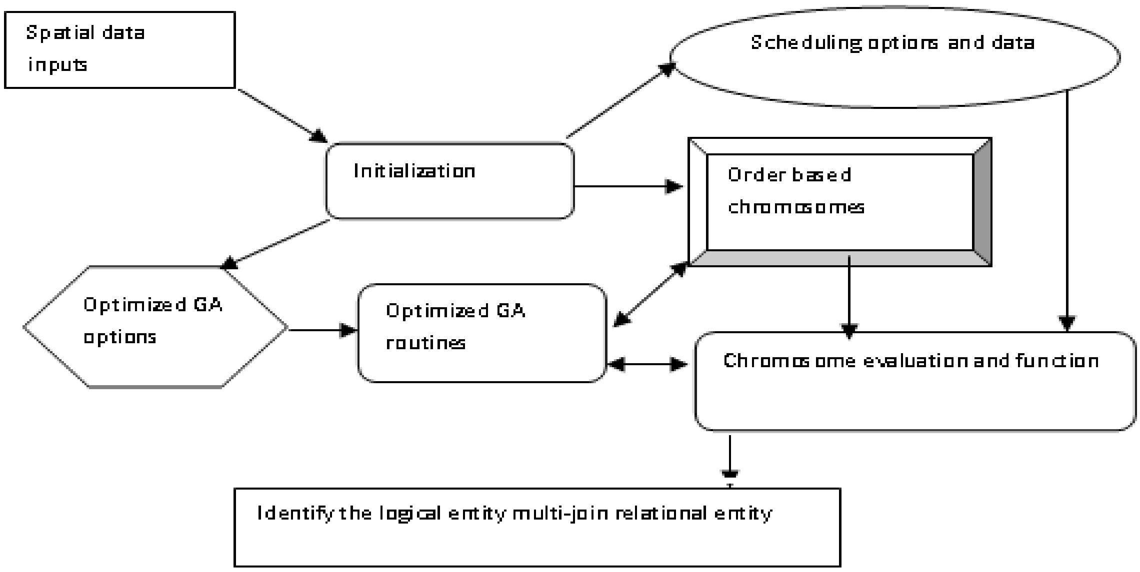


Figure 2.1 Data flow diagram of the optimized GA

From the Fig 3.2, it is noted that the initialization is done with the spatial data inputs accessed from the data warehouse. The initialization is done with an optimized order based chromosomes derived from the optimized GA options. The derived chromosomes evaluation is done with the fitness value generated. After thye chromosome evaluation is done with an optimized crossover and mutation, the logical multi-join relationship entity is identified. The algorithm below (fig 3.3) described the optimized multi-join relation with spatial data using optimized GA.

```
Input: Spatial data, Data warehouse
Output: best multi-join over spatial data extraction and transformation
Step 1: Evaluate the sets of candidate positive and negative terms
Step 2: Create the population OPop and initialize each chromosome
Step1: Create a new population (relational table)  NPop // Selection
Step2: Repeat
Step3:      For each chromosome (table) i in NPop do
Step4:                  Evaluate behavioral fitness f (i)
Step5:          Evaluate p[i]
```

```
Step6:                  Broadcast f (i) to all neighborhood of i
Step7:                  Based on fitness f,
Step8:                      Select the chromosomes (relational tables) from NPop
Step9:                  Form a set of group of individuals arbitrarily
Step10: Until the population variance in NPop is small
Step11: End
Step 12: Repeat
Step 13:   Estimate the size of NPop (New relational table)
Step 14:   Estimate the size of OPop (Existing relational table)
Step 15:  Compare the size of NewP with OldP
Step 16:   If Size(OldP)>Size(NewP)
Step 17:                Choose the parents' chromosomes (relational tables) in oldP
Step 18:                Estimate the optimized crossover probability of parents'
Step 19:                Evaluate a new variant
Step 20: End If
Step 21: Until the size of OPop equals with its values// optimized
                                                          crossover
Step 22: Produce kd1, kd2 through optimized crossover (parent1, parent2)
Step 23: apply optimized mutation, i.e.,   // optimized mutation
             kd1 = Optmut(kd1) and kd2 = Optmut(kd2)
Step 24: add kd1 and kd2 to newPop
Step 25: If OPop = NPop
Step 26: Select the best chromosome C in OPop;
Step 27: End If
Step 28: End
```

Figure 3 Optimized GA algorithm for identifying the logical relational entity with spatial data

From the Fig 3.3, a new population is formed by optimized GA like optimized mutation, optimized crossover and by combination of valuation system by selecting the primary population and the distribution of population by identifying the number of relational table presents. The behavioral fitness is computed by

$$FITNESS = \prod_{p=1}^{N_P} \{ \prod_{T=1}^{N_T} P_V (C_{P,T+1}) \} \ldots\ldots\ldots\ldots \text{(eqn 1)}$$

Where

 

$N_p$ :pixelnumber
$N_T$ :temporalslicenumber,
$C_{P,T}$ :class of the Pth pixel at the T time

Based on generation of new population, the probability of doing optimized crossover operation is done by choosing two relational tables among the probing population. The two chosen relational tables are optimally crossed with each other, for creating two new chromosomes or new child who is locate amongst the population. After completing the optimized crossover probability, compare the optimized mutation value of each group individuals and identify the optimized mutation value and the multi-join operation for those relational tables are efficiently performed by extracting the spatial data extraction and transformation.

## EXPERIMENTAL EVALUATION

The proposed optimized genetic algorithm for logical multi-join relational entity (OGALR) in active data warehouse is implemented in oracle. The experimental evaluation is conducted efficiently here for identifying the performance of the proposed OGALR for spatial data extraction and transformation. The effectiveness of the proposed optimized genetic algorithm for multi-join operation with spatial state of data is estimated with housing dataset derived from UCI repository with varying characteristics. The housing dataset consists of 506 instances with 14 attributes, but here we have taken 10 set of attributes. The table below shows the dataset which has been used for estimating the performance of the proposed OGALR.

Table 1 Housing data set

| CRIM | ZN | INDUS | CHAS | NOX | RM | AGE | DIS | RAD | TAX |
|---|---|---|---|---|---|---|---|---|---|
| 0.00632 | 18.00 | 2.310 | 0 | 0.5380 | 6.5750 | 65.20 | 4.0900 | 1 | 296.0 |
| 0.02731 | 0.00 | 7.070 | 0 | 0.4690 | 6.4210 | 78.90 | 4.9671 | 2 | 242.0 |
| 0.0279 | 0.00 | 7.070 | 0 | 0.4690 | 7.1850 | 61.10 | 4.9671 | 2 | 242.0 |
| 0.03237 | 0.00 | 2.180 | 0 | 0.4580 | 6.9980 | 45.80 | 6.0622 | 3 | 222.0 |
| .06905 | 0.00 | 2.180 | 0 | 0.4580 | 7.1470 | 54.20 | 6.0622 | 3 | 222.0 |
| 0.02985 | 0.00 | 2.180 | 0 | 0.4580 | 6.4300 | 58.70 | 6.0622 | 3 | 222.0 |
| 0.08829 | 12.50 | 7.870 | 0 | 0.5240 | 6.0120 | 66.60 | 5.5605 | 5 | 311.0 |
| 0.14455 | 12.50 | 7.870 | 0 | 0.5240 | 6.1720 | 96.10 | 5.9505 | 5 | 311.0 |
| 0.21124 | 12.50 | 7.870 | 0 | 0.5240 | 5.6310 | 100.00 | 6.0821 | 5 | 311.0 |
| 0.17004 | 12.50 | 7.870 | 0 | 0.5240 | 6.0040 | 85.90 | 6.5921 | 5 | 311.0 |

The above table (table 1) describes the dataset used for an experimentation to estimate the performance of the proposed optimized genetic algorithm for logical multi-join relational entity (OGALR) in active data warehouse. The descriptions of the specified attributes are

RIM (per capita crime rate by town), ZN (proportion of residential land zoned for lots over)25,000 sq.ft., INDUS (proportion of non-retail business acres per town), CHAS (Charles River dummy variable (= 1 if tract bounds river; 0 otherwise), NOX (nitric oxides concentration (parts per 10 million), RM (average number of rooms per dwelling), AGE (proportion of owner-occupied units built prior to 1940), DIS (weighted distances to five Boston employment centres), RAD (index of accessibility to radial highways), TAX (full-value property-tax rate per $10,000).

The proposed optimized genetic algorithm which has been applied to perform the multi-join operation in active data warehouse, improves the efficiency of multi-join operation in the active data warehouse. The proposed OGALR performed the multi-join operation using optimized GA to obtain the optimality based on the logical multi-join relational entity of the spatial state of data extraction and transformation. The performance of the proposed OGALR is measured in terms of

i) Multi-join Execution time,
ii) Optimal Query time
iii) Multi-joined attributes formation

**Execution time (multi-join)** is the total time taken by the proposed OGALR for processing the given amount of query at a particular interval of time t. The execution time is calculated as follows,

ET = KNO at time t ……. (eqn 1)

Where
ET - Execution time
K - Constant
N – Number of queries
O – No. of steps taken to process the query at time t

**Optimal Query time** is based on the processing time of the query using the proposed OGALR. The process of the proposed OGALR is experimented and measured with the above metrics.

## RESULTS AND DISCUSSION

When compared to an existing genetic algorithm used for multi-join operation in an Active Data Warehouse, the proposed optimized genetic algorithm for logical multi-join relational entity (OGALR) is effective in terms of query processing, execution time and operational overhead of the active data in a reliable manner. To perform the multi-join operation with spatial state of data, here we are using an optimized genetic algorithm to provide an efficient multi-join operation for the relational tables present in the active data warehouse. The below table and graph describes the performance of the proposed optimized genetic algorithm for logical multi-join relational entity (OGALR) in an active data warehouse.

Table 5.1 No. of tables vs. Execution time

| No. of tables (T) | Multi-join Execution time (sec) | |
|---|---|---|
| | Proposed OGALR | Existing GA for multi-join |
| 10 | 8 | 12 |
| 20 | 15 | 20 |
| 30 | 20 | 32 |
| 40 | 24 | 45 |
| 50 | 28 | 57 |

 

Table 5.1 described the multi-join execution time taken by the proposed OGALR from the tables present in the active data warehouse. The outcome of the proposed OGALR is compared with an existing GA for multi-join operation.

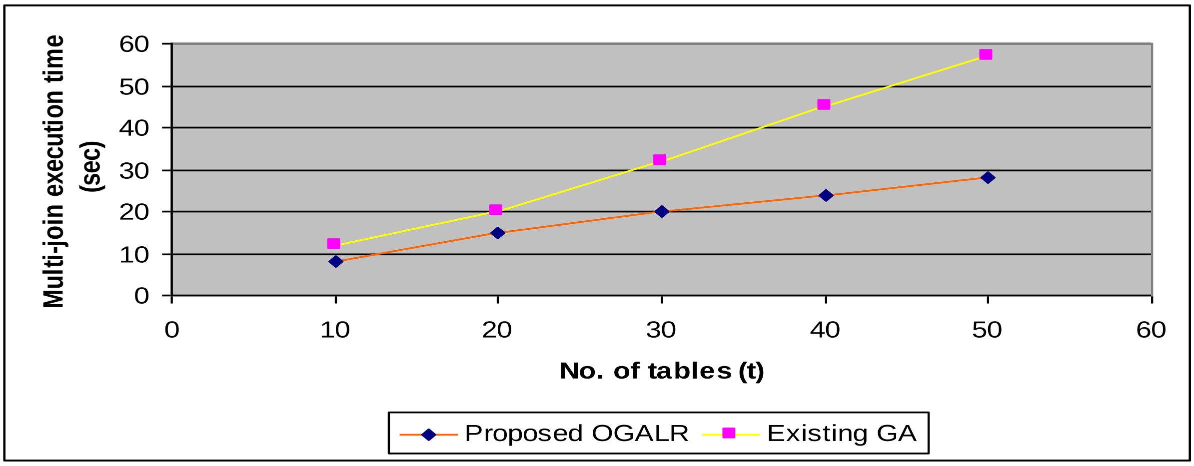


Figure 4 No. of tables vs. Execution time

Fig 5.1 describes the execution time taken to perform the multi-join operation among the relational tables present in the active data warehouse. The multi-join operation is done based on number of attributes present in the tables extracted from the data warehouse. The multi-join operation done using the proposed OGALR consumes less execution time compared to an existing GA technique. The multi-join operation execution time is measured in terms of seconds (secs). Since the proposed used optimized GA for multi-join operation, the attributes present in the table are easily joined in a less interval of time. Compared to an existing GA technique, the proposed OGALR performed the multi-join operation in a less execution time with a variance of 30-40% low.

Table 5.2 No. of tables vs. No. of multi-joined attributes created

| No. of tables | No. of multi-joined attributes created | |
|---|---|---|
| | Proposed OGALR | Existing GA for multi-join |
| 5 | 7 | 2 |
| 10 | 9 | 5 |
| 15 | 13 | 9 |
| 20 | 14 | 12 |
| 25 | 16 | 10 |

Table 5.2 described the process of forming the multi-join attributes from the set of relational tables present in the active data warehouse. The outcome of the proposed OGALR is compared with an existing GA for multi-join operation.

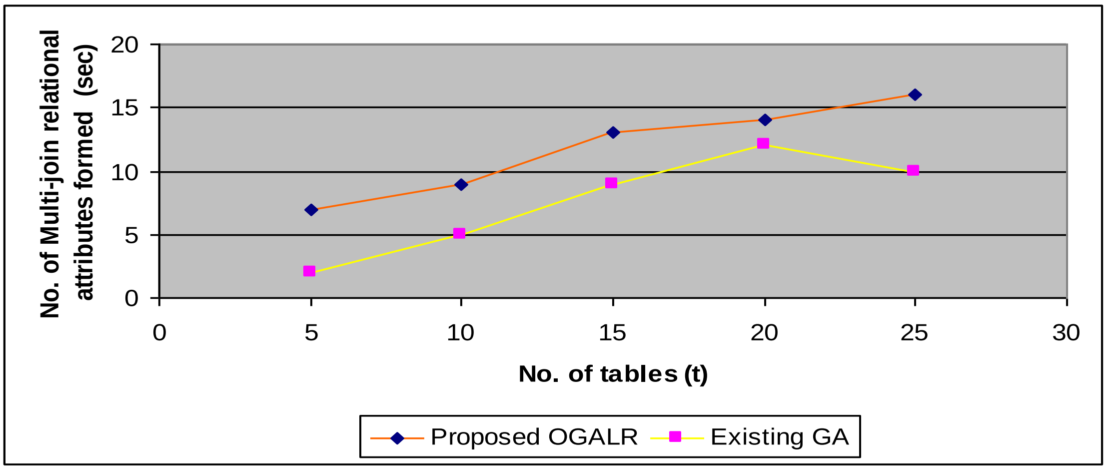


Figure 4.1 No. of tables vs. No. of multi-joined attributes created

Fig 5.2 describes the process of creating the multi-join attributes from the given number of tables present in the active data warehouse. The multi-join operation is done based on number of attributes present in the tables extracted from the data warehouse. The multi-join operation done using the proposed OGALR provides better logical multi-join relativity compared to an existing GA technique. Since the proposed used optimized GA for multi-join operation, the attributes present in the table are easily joined in a less interval of time. Compared to an existing GA technique, the proposed OGALR performed the multi-joined attributes in a less execution time with a variance of 20-30% high.

Table 5.3 No. of multi-join attributes vs. optimal query time

| No. of multi-joined attributes | Optimal query time (m/s) | |
|---|---|---|
| | Proposed OGALR | Existing GA for multi-join |
| 5 | 5 | 8 |
| 10 | 11 | 16 |
| 15 | 17 | 22 |
| 20 | 20 | 27 |
| 25 | 23 | 34 |

Table 5.3 described the process of accessing the attributes by the given type of query present in the active data warehouse. The outcome of the proposed OGALR is compared with an existing GA for multi-join operation.

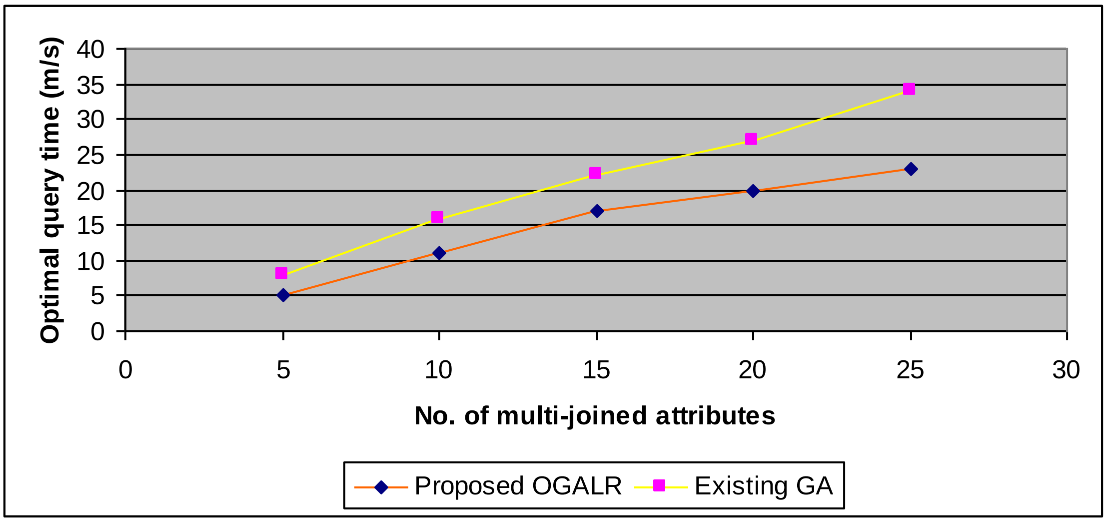


Figure 4.2 No. of multi-join attributes vs. optimal query time

Fig 5.3 describes the optimal query time to perform the multi-join operation among the attributes present in the tables. Based on the number of attributes present in the tables, the given queries have been processed with spatial state of data extraction and transformation. The optimal query processing time done using the proposed OGALR consumes less time compared to an existing GA technique. The optimal query processing time is measured in terms of milliseconds (m/s). Since the proposed used optimized GA for multi-join operation, the attributes present in the table are easily joined in a less interval of time. Compared to an existing GA technique, the proposed OGALR performed the optimal query processing in a less execution time with a variance of 20-30% low.

Based on the tables present in the active data warehouse, the multi-join operation is efficiently done to the attributes present in the table using an optimized GA technique. The optimized GA processed the query based on the multi-joined attributes derived from the tables and accessed in less interval of time. The optimization is done efficiently based on spatial state of data transformation and execution.

## CONCLUSION

The work presented here efficiently used an optimized Genetic Algorithm technique for multiple relations of joins being generated by streams from different direction in active data warehousing retrieval of data based on multiple queries. It has been evolved an optimized GA technique to have a more cohesive multi-relation joins of the stream tuples. An efficient OGALR technique is adapted in this work to aggregate the multiple queries efficiently to arrive more finite data extraction with optimal relational data items. Optimization is based on logical entity relation formed with spatial state of data extraction and transformation. Experimental simulations are carried out to evaluate the performance of proposed OGALR with housing data set derived from UCI repository. The performance of the proposed OGALR efficiently done the multi-join relation for the stream tuples present in the active data warehouse efficiently and the efficiency achieves 85% for multi-join relation operation.

## REFERENCES


[1]. Jie Liu, Wilson Wang, Farid Golnaraghi, "A GA-based Feature Optimization Technique for Bearing Fault Diagnostics", IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.9, September 2007

[2]. Chao-Lung Chiang_,"Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels", IEEE Transactions on Power Systems, 2005

[3]. dos Santos, A.G. , Mateus, G.R. , "General hybrid column generation algorithm for crew scheduling problems using genetic algorithm", IEEE Congress on Evolutionary Computation, 2009. CEC '09.

[4]. Hyunjung Lee , Byonghwa Oh et. Al., „Distributed genetic algorithm using automated adaptive migration", IEEE Congress on Evolutionary Computation, 2009. CEC '09.

[5]. Rodriguez, M.A._, Jarur, M.C. , "A genetic algorithm for searching spatial configurations", IEEE Transactions on Evolutionary Computation, 2005

[6]. Seunghyun Oh_et. Al., "Automatic Reproduction of a Genius Algorithm: Strassen's Algorithm Revisited by Genetic Search", IEEE Transactions on Evolutionary Computation, 2010

[7]. Jeurissen, R._et. Al., "Optimized index tracking using a hybrid genetic algorithm", (IEEE World Congress on Computational Intelligence). IEEE Congress on Evolutionary Computation, 2008. CEC 2008.

[8]. Golab, L., "Scalable Scheduling of Updates in Streaming Data Warehouses", IEEE Transactions on Knowledge and Data Engineering, 10 February 2011

[9]. Neoklis Polyzotis, Spiros Skiadopoulos et. Al., 'Meshing Streaming Updates with Persistent Data in an Active Data Warehouse', IEEE TRANSACTIONS ON KNOWLEDGE

AND DATA ENGINEERING, VOL. 20, NO. 7, JULY 2008

[10]. P. Delias, A. Doulamis, and N. Matsatsinis, "A Joint Optimization Algorithm for Dispatching Tasks in Agent-Based Workflow Management Systems," Proc. 10th Int'l Conf. Enterprise Information Systems (ICEIS '08), J. Cordeiro and J. Filipe, eds., pp. 199-206, 2008

**Short Bio Data for the Authors**

**Mr. K. Paramasivam** obtained M.Sc., Computer Science from P.S.G College, Bharathiyar University, Coimbatore, Tamil Nadu, India, in 1999, and M.Phil., Computer Science from Manonmaniam Sundaranar University, Thirunelveli, Tamil Nadu, India in 2003. He was working as IT analyst, in Department of Software development , **TATA Consultancy Services** ,Bangalore, Karnadaka, India. Currently he is working as Lead Engineer in Department of Software development, **Razorsight Software Pvt Ltd** (US Based Company) , Bangalore, Karnadaka, India. He is pursing Ph.D in Data warehouse.

**Dr. C. Chandrasekar** completed his Ph.D in Periyar University, Salem at 2006. He worked as Head, Department of Computer Applications at K.S.R. College of Engineering, Tiruchengode, Tamil Nadu, India. Currently he is working as Associate Professor in the Department of Computer Science at Periyar University, Salem, Tamil Nadu. His research interest includes Mobile computing, Networks, Image processing and Data mining. He is a senior member of ISTE and CSI.