A Review of Heuristics for Optimal Page Access Sequence with a Model for Spatial Queries

Dr.Chandra.E
Director
Department of Computer Science
DJ Academy for Managerial Excellence, Coimbatore

Anuradha.V.P
Research Scholar
Karpagam University
Coimbatore
+919486032409
anu.murug1991@gmail.com

***Abstract* - Spatial Database Systems are very large databases requiring storage in disk devices. The join process between two relations that is based on spatial nature of the attributes needs repeated and redundant disk accesses. This necessitates the implementation of join index which accelerates the join processing between two relations and it incurs high over head. This work studies the heuristics that are being in use for join processing from the context of optimal page access of disk I/O and that of spatial join processing with join index. The study presents a summary of heuristics that are implemented for join processing of spatial relations. Spatial-Queries modeled with spatial statistical measures provide an indication of spatial clusters. This model with an appropriate clustering algorithm identifies clusters that would require page re-accesses. An Optimal Page Access Sequence can reduce the page re-accesses and thus bring down the I/O cost of the join process. The work evaluates the performance of the statistical measure Moran's I to model spatial queries and identified to have a better performance than the existing measure Ripley's K function.**

Keywords – Spatial Database Management Systems, Spatial join, Join index, Join processing, Heuristics, Spatial statistical measure, Moran's I, Optimal page access sequence.

I. INTRODUCTION

Spatial Database Management Systems (SDBMS) are very large databases requiring vast amount of storage spaces. Even though main memory capacity has been increased by several giga bytes over the years, SDBMS needs secondary and tertiary storage spaces. In addition to this space constraint, there is a lack of ordering of data in SDBMS that is essential to preserve proximity between spatial objects. This constraint in spatial proximity leads to increase in the number of disk accesses. Spatial multidimensional indexes, which are the outcomes of the research in SDBMS, have found their use in a wide range of applications [16]. The retrieval of information from the relations present in an SDBMS depends upon the processing of such multidimensional indexes. Such a scenario requires the support for an additional operator termed Spatial Join in the query language.

The joining of two relations is based on their spatial attributes and this process incurs a high overhead. Hence the implementation of accelerators in the join operation becomes significant. The function of spatial join operator is based on some spatial join predicate, just like a relational join predicate. The spatial join predicate can be a geometric relationship between two relations, such as enclosure, intersection, overlap etc. Spatial indices provide acceleration to spatial queries, thereby helping in the effective execution of such queries.

Many of the queries based on spatial attributes make use of a spatial neighborhood matrix known as W in spatial statistics. These queries have to compute W Matrix several times for different neighborhoods. Such queries are called as Spatial-Queries [13]. The processing of joining queries in spatial database management systems has been enhanced through spatial index structures such as B-Tree and R-Tree index families. The SDBMSs are required to perform repeated computation of the W-Matrix with in a limited time frame. The specifications in their functionalities do not provide for the processing of Spatial-Query operations. It is claimed that SDBMS should support a self-join index [13].

The purpose of this paper is to incorporate an Optimal Page Access Sequence with that of a spatial join index structure and thereby attempts to reduce the number of disk or page re-accesses.

Research related to the spatial index structure needed for join processing in Spatial Database Management Systems are being carried out in a separate stream from that of the evaluation of optimal page access sequence for disk I/O processes. Both the kind of works aim to bring down the cost I/O operations of disk accesses. An analysis of wide range of techniques for selection of spatial indexing structure has been done in past literature [1,8,9,10,11,13,15]. Indexing structures for multidimensional data such as R-Trees, R*-Trees, Quad Tree, k-d Tree, etc have been used for the determination of spatial proximity for a single neighbourhood relationship have widely implemented [1,9,12]. Extensive research work has gone in to the selection of spatial index structure required for join processing. A formal definition for join indices was proposed for relational join operations [8]. Spatial join index as a data structure was implemented with grid files [15]. Improvement in the computational efficiency of spatial indices was performed with the construction of spatial index using Generalization trees [8].

A major issue faced by existing SDBMS tools to support several spatial indices is that the choice of a spatial index type should satisfy such requirements as optimal I/O accesses, optimization of query processing, concurrency control, and recovery plans. Spatial database systems are very large databases that are updated infrequently. Join indices that form as basis for spatial join indices perform the join process to reduce query response time in online query processing. W-queries can be modeled with Moran' I, Geary's C and 'G' statistic [4]. W-queries perform self-join computation a number of times, in which case the join index is a neighbourhood graph instead of bi-partite graphs of the previous implementations [13].

Previous works related to optimal page access sequence needed for the proximity constrained SDBMS has been dealt with in the literature [6]. Page access sequence in case of disk accesses requires effective scheduling processes and algorithms. Detailed work to address this issue has been conducted in past literatures [2].

This work makes an analytical study of the heuristics to generate an optimal page access sequence in join processing. This paper focuses on the implementation of a spatial index type for a specified set of operations over a given study area. This work aims at the efficient execution of spatial neighbourhood queries over multiple spatial instances of an area under study. This work models spatial queries with a spatial statistical measure and evaluates its performance with sample data sets.

The rest of the paper is organized as follows. Section 2 explains preliminaries, the join index. Section 3 presents an analytical study of the heuristics for optimality of page access sequence in conjunction with a spatial index type structure. Section 4 consists of the model for spatial queries and its performance evaluation with a pseudo code. Section 5 concludes the work with observations in the context of heuristics for optimal page Access Sequence and model for Spatial-Queries.

II. PRELIMINARIES

In this section the basic concept of join index is described for a sample study region consisting of spatial objects, Buildings and Parks.

A. Join Index

The concept of an index termed join index that was implemented as an accelerator for performing join operations between two relations has provided basis for spatial joins. The join index is a pre-computed structure that indicates which tuples from one relation will match tuples of the other relation based on some join predicate [8]. It was established that these indices can be used by the query optimizer in a general query, and can also be utilized in effective implementation of recursive join queries. With theses indices the maintenance of referential integrity constraints can be done with very small overhead.

As an example, consider two data sets D_B and D_P that cover some common area. D_B contains all buildings in the area , and D_P contains all parks in the area. Assuming that R-tree indices have been constructed for both D_B and D_P, consider the following join operations:

Ql: Find all buildings that overlap with a park

Q2: Find all government-owned buildings that

overlap with a park.

Figure. 1 illustrates the locations of buildings (shown as circles) that overlap parks (shown as polygons bounded within rectangles) in a given study region. R is the radius within which the query Q1 is evaluated. Table I. Represents the join index of the relations Buildings and Park.

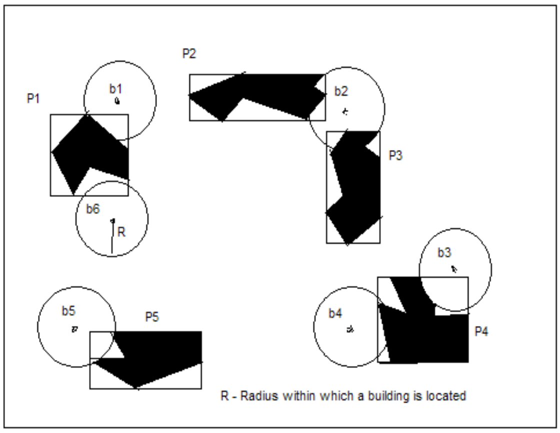


Figure. 1 Location – Buildings and Parks

Table I. Join Index

Building_id	Park_id
b1	P1
b2	P2
b2	P3
b3	P4
b4	P4
b5	P5
b6	P1

III. HEURISTICS FOR OPTIMAL PAGE ACCESS SEQUENCE IN JOIN PROCESSING

Heuristics required for Optimal Page Access Sequence that have to be incorporated with join processing to reduce I/O cost have been studied in this section. The heuristics have been modeled with data structures: Page Connectivity Graph, Page-Pair Graph, Bi-Partite Graph and Neighbourhood Graph. Each of the heuristics is reviewed in brief in this section.

A.Heuristics with Page Connectivity Graph

The evaluation of query process requires the join operations to be performed between two relations. Heuristics for optimal page access sequence and optimal block access sequence are modeled with connectivity graphs. The heuristics models Page Access Sequence with a buffer size constraint with various semantics. The models are based on the joining operations between pages, blocks and tuples of the relations R and S [6]. The subsequent analysis of the performance of the models leads to the conclusion that the heuristics are NP-hard problems [3]. The heuristic for optimal page access sequence is stated as,

Heuristic [6]:

Step 1: Initialize the buffer to empty.

Step 2: Add a node, say *X* , at random to the buffer.

Step 3: Add to the buffer all the nodes which are adjacent to node *X*.

Step 4: Perform join among the nodes (data pages) in the buffer.

Step *5:* Remove from the buffer the nodes which have all of their adjacent nodes in the buffer.

Step 6: If the buffer is empty then STOP. Otherwise let *X* be a random node in the buffer. Go to Step 3.

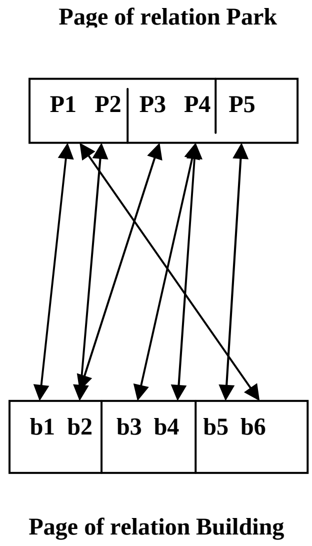


Figure 2. Page Connectivity Graph

Table II. Join Adjacency List

Node X	Adjacent nodes
b1	b2, P1, P2

Figure 2. illustrates the page connectivity graph that represents the join index of the two relations. Table II. is the page access sequence for the sample node b1 of the page connectivity graph.

B. Heuristics for Effective Page Access Schedule with Page-Pair Graph

Heuristics for page access sequence has been evaluated for a buffer of optimal size. The issue of scheduling the page accesses is addressed in the correct semantics in this heuristics [2]. A procedure to compute the join of *X* and *Y* is stated as follows in a two-step algorithm: 1) The relevant indices on the two relations considered for joining operations to obtain a list of pairs of data page identifiers (*x*, *y*) where page *x* contains some tuple which needs to join with some other tuple in page *y*. 2) The page-pair information from Step 1 can be utilized to schedule an efficient page access sequence to fetch the data pages. Such a procedure can be used when join indexes (consisting of a list of pairs of related tuple/ object identifiers) are used to improve the execution rate of join computation.

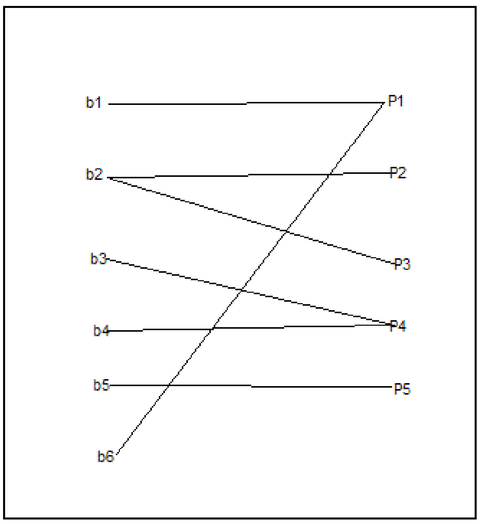


Figure 3. Page-Pair Graph for Join Index

Figure 3. shows the page-pair connectivity graph for join of relations Building and Park. The buffer used in the join operations should be utilized in such a way that it contains pages up to its optimum capacity, in order to limit page re-accesses. This requirement produces issues that are given as follows:

1) What is the optimal page access sequence such that the join can be computed without any page re-accesses using the minimum number of buffer pages?

2) Given a fixed number of buffer pages, what is the optimal page access sequence such that the join can be computed with the minimum number of page re-accesses?

These issues are addressed with appropriate heuristics with the pages being represented by an undirected page-pair graph. The heuristics are developed by modeling a page access sequence as a concatenation of segments. The expected properties of a page access sequence are achieved with the heuristics, which can be stated as, (1) It should not contain premature page accesses, (2) It should be a minimal page access sequence. The heuristic [2], addresses the OPAS (Optimal Page Access Sequence) problem and produces page access sequences satisfying both the properties. The optimum page access sequence for the data set would be (b1,P1,b6,P2,b2,P3,b3,P4,b4,b5,P5).

C. Heuristics for Join Processing with Spatial Clustering Approach

Join processing with spatial clustering asymmetric clustering or incremental clustering approach. The incremental clustering approach [14] has two heuristics, 1) to reduce the buffer size required for the join process and 2) to perform join with a fixed sized buffer. The join index is modeled either as a bipartite graph or a pair-page graph. The number of re-accesses to the buffer should be limited, which gets violated due to the fixed number of pages stored in the buffer. During join processing, the page access sequence should be evaluated to reduce the number of page re-accesses. The solutions achieved in previous literatures have proved them to be NP-hard problems [3]. Figure 4. is the adjacency matrix for the join index in Table 1. of the two relations.

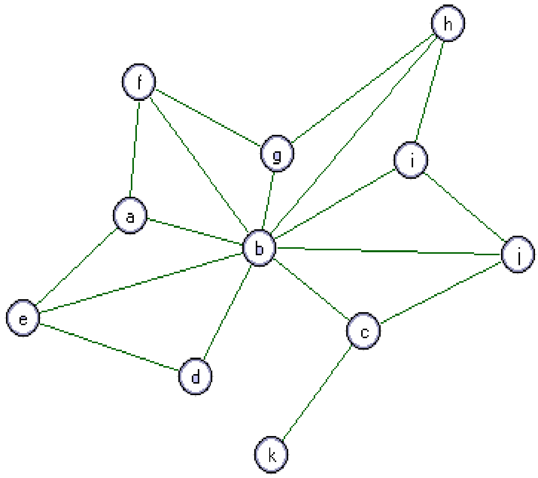
$$\begin{bmatrix} & \boldsymbol{P1} & \boldsymbol{P2} & \boldsymbol{P3} & \boldsymbol{P4} & \boldsymbol{P5} \\ \boldsymbol{b1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{b2} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{b3} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \boldsymbol{b4} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \boldsymbol{b5} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ \boldsymbol{b6} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

Figure 4. Adjacency matrix for the Join Index

A different set of heuristics is put forth in a literature [18] that is based on clustering method to group the pages of a single relation. One of the heuristics generalizes sorting based methods and the other generalizes incremental clustering method. Based on spatial clustering two methods namely, Asymmetric method and Symmetric method are implemented over two relations. The attribute of one of the relations is used as a key for the join processing in asymmetric method, whereas in the symmetric method the keys of attributes of either one of the relations are implemented in the join process evaluation. The bipartite graph and the page-connectivity graph are representations of join indexes that can be used in the context of spatial data ware houses.

D. Heuristics for Join Processing with Spatial Clustering and a Self-Join Index

Join processing with spatial clustering approach is implemented with a self-join index [13]. In this approach, a number of neighbourhood relationships is generated for the spatial relation under consideration. For each of the neighbourhood relation, appropriate clustering algorithm is applied to determine clusters in the dataset. The join processing is carried out with two algorithms, 1) algorithm to determine the presence of clusters, 2) algorithm to identify the clusters. The Spatial-Queries are modeled with spatial statistics to effectively reduce the I/O cost.the self-join index is better represented by the data structure, neighbourhood graph.



(a)

	a	b	c	d	e	f	g	h	i	j	k
a	0	1	0	0	1	1	0	0	0	0	0
b	0	0	1	1	1	0	1	1	1	0	0
c	0	1	0	0	0	0	0	0	0	1	1
d	0	1	0	0	1	0	0	0	0	0	0
e	1	1	0	1	0	0	0	0	0	0	0
f	1	0	0	0	0	0	1	0	0	0	0
g	1	1	0	0	0	1	0	0	0	0	0
h	0	1	0	0	0	0	0	0	1	0	0
i	0	1	0	0	0	0	0	1	0	1	0
j	0	0	1	0	0	0	0	0	1	0	0
k	0	0	1	0	0	0	0	0	0	0	0

(b)

Figure 5(a). Neighbourhood Graph and Figure1(b).W-matrix for the neighbourhood graph

Figure.5(a) shows the neighbourhood graph for a sample dataset of a crime data warehouse. A neighbourhood graph [5] is defined in [6] as a graph $\boldsymbol{G^{Db}_{neighbour}} = (\boldsymbol{N}, \boldsymbol{E})$ with nodes $\boldsymbol{N} = \boldsymbol{DB}$ and edges $\boldsymbol{E} \subseteq \boldsymbol{N} \times \boldsymbol{N}$ where an edge $\boldsymbol{e} = (\boldsymbol{n_1}, \boldsymbol{n_2})$ exists iff $\boldsymbol{neighbour}(\boldsymbol{n_1}, \boldsymbol{n_2})$ holds. The nodes a,b,c,…,j,k in the above neighbourhood graph represent spatial objects of the sample dataset. Figure. 5(b) is the associated W-matrix for the neighbourhood graph. Spatial statistics defines W-Matrix as a matrix representation of space [7]. It provides an indication of the adjacency of spatial objects. Also it illustrates the degree of spatial interaction among spatial objects of a spatial dataset. The value 1 in the matrix indicates that the corresponding objects are adjacent to each other and a 0 value indicates the absence of adjacency between the spatial objects. The neighbourhood graph takes the characteristics of the regular operations of a graph data structure [3].

IV. MODELING SPATIAL-QUERIES WITH SPATIAL STATISTICS

In this work modeling of Spatial-Queries is performed with the implementation of spatial statistical measures such as Moran' I, Geary' s C and 'G' statistic characterize their computational structure. In spatial statistics, the W-Matrix is a matrix based representation of space and a measure of the adjacency, proximity, distance or level of spatial interaction between spatial instances [7]. The Spatial-queries are evaluated with a set of operations that are relevant with the neighbourhood graph representation of the self-join index.

A. Spatial-Queries and Moran's I

Query : Does the dataset contain spatial clusters?

Spatial-Queries are executed by modeling them with a suitable statistical measure. The Moran I index is a statistical measure that provides an indication of the presence of clusters in the given spatial datasets. The Moran *I* index, that is global index of spatial statistics can be disaggregated to provide a series of local indices. The standard formulation is stated as:

$$I = \frac{1}{p} \frac{\sum_i \sum_j w_{ij} (z_i - z')(z_j - z')}{\sum_i (z_i - z')^2}, where$$

$$p = \sum_i \sum_j w_{ij} / n$$

For a given study region the individual components, or Local Indicators of Spatial Association (LISA) provide an indication of spatial clustering. Another statistic, "G", due to Getis and Ord (1992, 1995) may also be used in this way.

B.Operations for Spatial-Queries

In the self join computation of the join process, the spatial index is represented as a neighborhood graph. In that context, the operations and terminology implemented in a spatial network literature is made use of in this evaluation of self join spatial index [16].

The functions that can be incorporated into the query processing algorithm may be stated as follows,

- get-neighbours-in-relationship(Ob_i,R): Identify the neighbours of a spatial object Ob_i.
- get-successors(Ob_i): Retrieve the successors of Ob_s.
- get-successor(Ob_i): Retrieve the farthest unreported successor of Ob_i.
- get-predecessors(Ob_i): Retrieve the predecessors of Ob_i.
- get-predecessor-of-successor(Ob_i): Retrieve the predecessors of successor of Ob_i.
- get-predecessors-of-successor(Ob_i): Retrieve the predecessors of successors of Ob_i.
- update-successors(Ob_i, <successors>): Un-colors all the successors of Ob_i.
- update-average edge weight(Ob_i): Update the average edge weight of a spatial object.

where,

R – Neighbourhood relation

Ob_i – Spatial object on which the functions are implemented

C.Evaluation of Spatial-Queries

The above specified set of operations may be used to evaluate the W-Queries. The aim of the evaluation process is minimization of the I/O costs of page accesses. This can be achieved with the implementation of a special type of spatial index named as self-join index and its variants namely self-join edge list index and self-join adjacency list index. The self-join index is defined as,

The set of W-Query operations makes use of a coloring heuristic to return the result of an evaluation. The observations that the maximum size of a page and the value of the Connectivity Residue Ratio (CRR) [18] affect the storing of the pages in secondary devices is considered for evaluation in this study. The connectivity clustering heuristic on the neighborhood graph can be used for the allocation of these spatial instances to disk pages.

4.3.1 Spatial-Query Processing Algorithms

In this section, the set of operations get-neighbors-in-relationship(), getsuccessors(), get predecessors(), get-successor(), getpredecessor(), get-predecessor-of-successor(), get-predecessorsof- successor(), update-average-edge-weight(), and updatesuccessors() are utilized in the design of the algorithms for W-Queries. The Queries under consideration are Moran’s I, Geary ‘C’ ‘G’ stastic

4.3.1.1 Evaluation of Moran’s I

Pseudo code:

Begin,

Consider each spatial instance in the given spatial dataset S_d.

Compute successor of i^{th} node

Store in array Successor[i]

Compute Moran’ I with the equation,

End.

Table III. illustrates the spatial join adjacency list when the spatial query is modeled with Moran’s I for cluster identification. Table IV. shows the results of the spatial query evaluation. The results show a better performance with Moran’s I as against the existing Ripley’s K function.

Table III. Spatial Join Adjacency List

Ob_i	**Adjacency List**
a	b,e,f
b	e,d,g,h,i,c
c	b,k,j
d	b,e
e	a,b,d
f	a,g
g	a,b,f
h	b,i
i	b,h,j
j	c,i
k	c

Table IV. Results of the Spatial Query Evaluation

Dataset size	**User response time in seconds**	
	RipleyK function	***Moran's I***
14852	93.78	67.509
4489	50.35	20.405
2290	20.612	10.409
1182	10.815	5.373

Figure 6. shows the comparison chart of the performance evaluation of the statistical measures Ripley's K function and Moran's I for varying data set sizes.

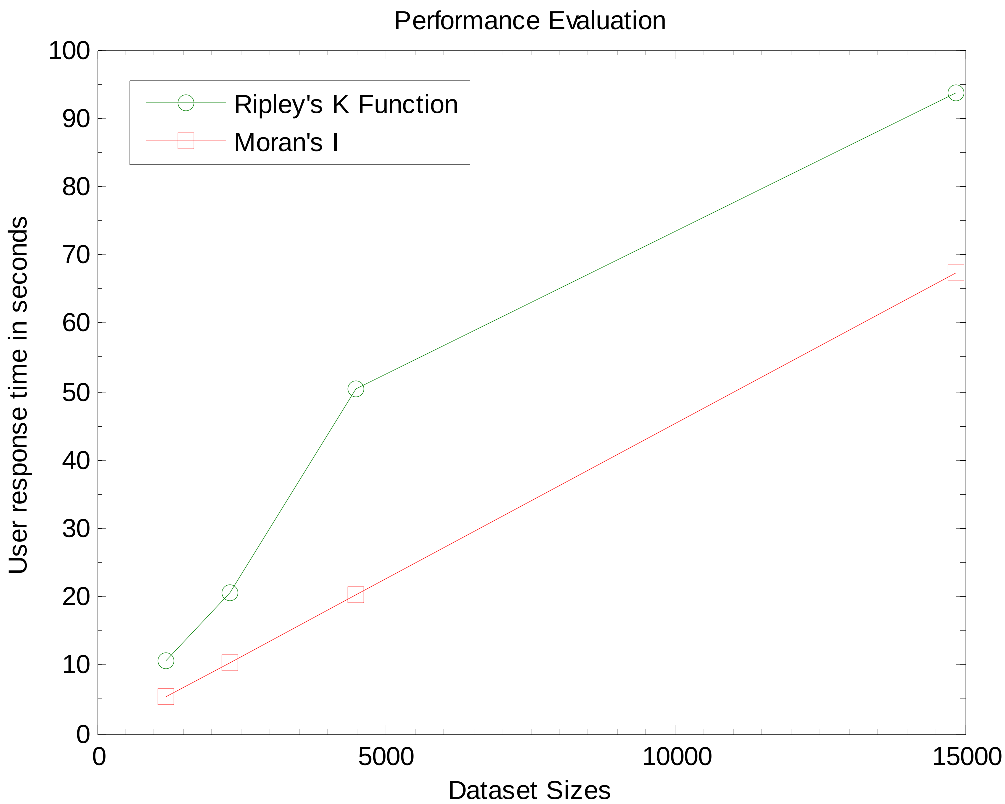


Figure 6. Comparison chart of the Performance Evaluation

V. CONCLUSION

The join processing of relations of an SDBMS using join indexes of spatial nature are constrained due to spatial proximity. Also there is lack of ordering in storage of pages in secondary storage devices due to the constraint. This drawback leads to page re-accesses, thereby increasing the I/O cost of disk accesses. This work has made a study of the existing literatures to determine the evaluation techniques for join processing that can overcome this constraint. Heuristics to generate an Optimal Page Access Sequence have been studied. Join index accelerates the join processing in SDBMS. The heuristics implement either a page-connectivity graph or a page-pair graph or a bi-partite graph as the data structure for the join index. Clustering algorithms are applied to the pages involved in the join process in order to reduce the processing time. The heuristics with the clustering algorithms along with Optimal Page Access Sequence generation bring down the I/O cost of disk accesses. Clustering algorithms such as incremental clustering, asymmetric clustering and symmetric clustering have been implemented by these heuristics. A more recent work identifies Self-Join index as a suitable join index for join processing in SDBMS as opposed to the on-the fly computation of join indexes. A neighbourhood graph is an appropriate representation for the self-join index. In this work, Spatial-queries are modeled with Moran's I index and evaluated with a set of operations as against the Ripley's k function of the previous work. The spatial statistical measure is evaluated with a sample dataset and it is identified to have better user response time for a self-join computation. The future work would have a model that incorporates the heuristics needed for optimal page access sequence. As clustering of the pages involved in the join process reduces the process time considerably, an enhanced and improvised clustering algorithm would be implemented with the chosen join index in future. In the future work, the join selectivity factor for the spatial join would be taken up in order to bring down the I/O cost further.

REFERENCES

[1] N. Beckmann, H.P. Kriegel, R. Schnieder and BB. Seeger, "The R*-tree: an efficient and robust access method for points and rectangles" SIGMOD Rec. , 19(2): 322-331, 1990.

[2] C.Y. Chan and B.C. Ooi, "Efficient Scheduling of Page Access in Index-Based Join Processing," IEEE Trans. Knowledge and Data Eng., vol. 9, no. 6, pp. 1005-1011, Nov./Dec. 1997.

[3] T. Cormen, C. Leiserson, and R. Rivest, "Introduction to Algorithms", The MIT Press, 1991.

[4] De Smith, Goodchild, Longley, "Geospatial Analysis A comprehensive guide", 2nd Edition, 2006-2008.

[5] Ester.M, Kriegel.H.P, Sander.J, "Algorithms and applications for Spatial Data Mining", Geographic Data Mining and Knowledge Discovery, Research Monographs in GIS, Taylor and Francis, 2001.

[6] F. Fotouhi and S. Pramanik, "Optimal Secondary Storage Access Sequence for Performing Relational Join," IEEE Trans. Knowledge and Data Eng., vol. 1, no. 3, pp.318-328, Sept. 1989.

[7] V. Gaede and O. Gunther, "Multidimensional access methods", ACM Comput. Surv., 30(2): 170-231,1998.

[8] O. Gunther, "Efficient Computation of Spatial Joins," Proc. Int'l Conf. Data Eng., 1993.

[9] A Guttman, "R Trees: a dynamic index structure for spatial searching". In SIGMOD'84: ,Proceedings of the 1984 ACM SIGMOD international conference on Management of data, pages 47-57, New York, NY, USA. 1984.ACM.

[10] E.H. Jacox and H.Samet, "Spatial Join Techniques", ACM Transactions on Database Systems., 32(1): 7, 2007.

[11] M. Lo and C.V. Ravishankar, "Spatial Joins Using Seeded Trees," Proc. 1994 ACM SIGMOD Int'l Conf. Management of Data, pp. 209- 220, 1994.

[12] G.Malcom, Microsoft SQL Server 2008, Delivering Location Intelligence with Spatial Data. SQL Server Technical Article. Microsoft Corporation, Aug 2007.

[13] P. Mohan, S.Shekhar, N. Levine, R.E.Wilson, B.George and M.Celik, "Should SDBMS. Support a Join Index?: A Case Study from CrimeStat" In Proc. of 16th ACM SIGSPATIAL International Conferrence on Advances in Geographic Information Systems (ACM GIS), California, USA, 2008.

[14] E.R. Omiecinski, "Heuristics for Join Processing Using Nonclustered Indexes," IEEE Trans. Software Eng., vol. 15, no. 1, pp. 18-25, Jan. 1989.

[15] D. Rotem. Spatial Join Indices. In Proceedings of the Seventh International Conference on Data Engineering, April 8-12, 1991, Kobe Japan, pages 500-509. IEEE Computer Society, 1991.

[16] S. Shekhar, S. Chawla, S. Ravada, A. Fetterer, X. Liu, and C.T. Lu, "Spatial Databases: Accomplishments and Research Needs," IEEE Trans. Knowledge and Data Eng., vol. 11, no. 1, pp. 45-55, Jan./Feb. 1999.

[17] S. Shekhar and D. R. Liu, "CCAM: A Connectivity-Clustered Access Method for Networks and Network Computations", IEEE Trans. on Knowledge and Data Engineering, Vol. 9, No. 1, Jan. 1997

[18] S.Shekhar, C.T. Lu, S.Chawla and S.Ravada, "Efficient Join-Index- Based Spatial Join Processing: A Clustering Approach", IEEE Trans. In Know. and Data Engineering 15(1), 2003.

AUTHORS PROFILE

Dr.E.Chandra received her B.Sc., from Bharathiar University, Coimbatore in 1992 and received M.Sc., from Avinashilingam University ,Coimbatore in 1994. She obtained her M.Phil., in the area of Neural Networks from Bharathiar University, in 1999. She obtained her PhD degree in the area of Speech recognition system from Alagappa University Karikudi in 2007. She has totally 14 yrs of experience in teaching including 6 months in the industry. At present she is working as Assistant Professor, Department of Computer Applications in D. J. Academy for Managerial Excellence, Coimbatore. She has published more than 30 research papers in National, International journals and conferences in India and abroad. She guided more than 40 M.Phil., research scholars. At present 12 M.Phil Scholars and 6 Ph.D Scholars are working under her guidance. She has delivered lectures to various colleges. She is a Board of studies member at RVS College. Her research interest lies in the area of Data Mining, Artificial intelligence, neural networks, speech recognition systems and fuzzy logic and Machine Learning Techniques. She is an active member and Life member of CSI, Society of Statistics and Computer Applications. Currently Management Committee member of CSI Coimbatore Chapter.

Anuradha.V.P received B.E. in Civil Engineering from Bharathiar University, Coimbatore in 1989. She received the Masters degree MCA in Computer Applications from Bharathiar University in 2004. She received MPhil degree in Computer Science in the area of Data Mining from Bharathiar University in 2008. She is currently doing research work for the PhD degree in Computer Science at Karpagam University under the guidance of Dr. E. Chandra. Her research interests include spatial data mining, algorithm design and analysis. She has presented a paper in IEEE IACC '09 (International Advance Computing Conference) in Patiala,India.