INTERNATIONAL JOURNAL OF RESEARCH IN COMMERCE, IT & MANAGEMENT

IJRCM

IJRCM

A Monthly Double-Blind Peer Reviewed (Refereed/Juried) Open Access International e-Journal - Included in the International Serial Directories

Indexed & Listed at:

Ulrich's Periodicals Directory ©, ProQuest, U.S.A., EBSCO Publishing, U.S.A., Cabell's Directories of Publishing Opportunities, U.S.A.,
Open J-Gage, India [link of the same is duly available at Inflibnet of University Grants Commission (U.G.C.)],
Index Copernicus Publishers Panel, Poland with IC Value of 5.09 & number of libraries all around the world.

Circulated all over the world & Google has verified that scholars of more than 3412 Cities in 173 countries/territories are visiting our journal on regular basis.

Ground Floor, Building No. 1041-C-1, Devi Bhawan Bazar, JAGADHRI – 135 003, Yamunanagar, Haryana, INDIA

http://ijrcm.org.in/

CONTENTS

CALL FOR MANUSCRIPTS

We invite unpublished novel, original, empirical and high quality research work pertaining to recent developments & practices in the areas of Computer Science & Applications; Commerce; Business; Finance; Marketing; Human Resource Management; General Management; Banking; Economics; Tourism Administration & Management; Education; Law; Library & Information Science; Defence & Strategic Studies; Electronic Science; Corporate Governance; Industrial Relations; and emerging paradigms in allied subjects like Accounting; Accounting Information Systems; Accounting Theory & Practice; Auditing; Behavioral Accounting; Behavioral Economics; Corporate Finance; Cost Accounting; Econometrics; Economic Development; Economic History; Financial Institutions & Markets; Financial Services; Fiscal Policy; Government & Non Profit Accounting; Industrial Organization; International Economics & Trade; International Finance; Macro Economics; Micro Economics; Rural Economics; Co-operation; Demography: Development Planning; Development Studies; Applied Economics; Development Economics; Business Economics; Monetary Policy; Public Policy Economics; Real Estate; Regional Economics; Political Science; Continuing Education; Labour Welfare; Philosophy; Psychology; Sociology; Tax Accounting; Advertising & Promotion Management; Management Information Systems (MIS); Business Law; Public Responsibility & Ethics; Communication; Direct Marketing; E-Commerce; Global Business; Health Care Administration; Labour Relations & Human Resource Management; Marketing Research; Marketing Theory & Applications; Non-Profit Organizations; Office Administration/Management; Operations Research/Statistics; Organizational Behavior & Theory; Organizational Development; Production/Operations; International Relations; Human Rights & Duties; Public Administration; Population Studies; Purchasing/Materials Management; Retailing; Sales/Selling; Services; Small Business Entrepreneurship; Strategic Management Policy; Technology/Innovation; Tourism & Hospitality; Transportation Distribution; Algorithms; Artificial Intelligence; Compilers & Translation; Computer Aided Design (CAD); Computer Aided Manufacturing; Computer Graphics; Computer Organization & Architecture; Database Structures & Systems; Discrete Structures; Internet; Management Information Systems; Modeling & Simulation; Neural Systems/Neural Networks; Numerical Analysis/Scientific Computing; Object Oriented Programming; Operating Systems; Programming Languages; Robotics; Symbolic & Formal Logic; Web Design and emerging paradigms in allied subjects.

Anybody can submit the **soft copy** of unpublished novel; original; empirical and high quality **research work/manuscript** ***anytime*** in ***M.S. Word format*** after preparing the same as per our **GUIDELINES FOR SUBMISSION**; at our email address i.e. infoijrcm@gmail.com or online by clicking the link **online submission** as given on our website (***FOR ONLINE SUBMISSION, CLICK HERE***).

GUIDELINES FOR SUBMISSION OF MANUSCRIPT

1. **COVERING LETTER FOR SUBMISSION:**

DATED: _____________

THE EDITOR
IJRCM

Subject: **SUBMISSION OF MANUSCRIPT IN THE AREA OF.**

(e.g. Finance/Marketing/HRM/General Management/Economics/Psychology/Law/Computer/IT/Engineering/Mathematics/other, please specify)

DEAR SIR/MADAM

Please find my submission of manuscript entitled '____' for possible publication in your journals.

I hereby affirm that the contents of this manuscript are original. Furthermore, it has neither been published elsewhere in any language fully or partly, nor is it under review for publication elsewhere.

I affirm that all the author (s) have seen and agreed to the submitted version of the manuscript and their inclusion of name (s) as co-author (s).

Also, if my/our manuscript is accepted, I/We agree to comply with the formalities as given on the website of the journal & you are free to publish our contribution in any of your journals.

NAME OF CORRESPONDING AUTHOR:
Designation:
Affiliation with full address, contact numbers & Pin Code:
Residential address with Pin Code:
Mobile Number (s):
Landline Number (s):
E-mail Address:
Alternate E-mail Address:

NOTES:
a) The whole manuscript is required to be in ***ONE MS WORD FILE*** only (pdf. version is liable to be rejected without any consideration), which will start from the covering letter, inside the manuscript.
b) The sender is required to mentionthe following in the **SUBJECT COLUMN** of the mail:
New Manuscript for Review in the area of (Finance/Marketing/HRM/General Management/Economics/Psychology/Law/Computer/IT/Engineering/Mathematics/other, please specify)
c) There is no need to give any text in the body of mail, except the cases where the author wishes to give any specific message w.r.t. to the manuscript.
d) The total size of the file containing the manuscript is required to be below **500 KB**.
e) Abstract alone will not be considered for review, and the author is required to submit the complete manuscript in the first instance.
f) The journal gives acknowledgement w.r.t. the receipt of every email and in case of non-receipt of acknowledgment from the journal, w.r.t. the submission of manuscript, within two days of submission, the corresponding author is required to demand for the same by sending separate mail to the journal.

2. **MANUSCRIPT TITLE**: The title of the paper should be in a 12 point Calibri Font. It should be bold typed, centered and fully capitalised.

3. **AUTHOR NAME (S) & AFFILIATIONS**: The author (s) **full name**, **designation**, **affiliation** (s), **address**, **mobile/landline numbers**, and **email/alternate email address** should be in italic & 11-point Calibri Font. It must be centered underneath the title.

4. **ABSTRACT**: Abstract should be in fully italicized text, not exceeding 250 words. The abstract must be informative and explain the background, aims, methods, results & conclusion in a single para. Abbreviations must be mentioned in full.

5. **KEYWORDS**: Abstract must be followed by a list of keywords, subject to the maximum of five. These should be arranged in alphabetic order separated by commas and full stops at the end.

6. **MANUSCRIPT**: Manuscript must be in ***BRITISH ENGLISH*** prepared on a standard A4 size ***PORTRAIT SETTING PAPER***. It must be prepared on a single space and single column with 1" margin set for top, bottom, left and right. It should be typed in 8 point Calibri Font with page numbers at the bottom and centre of every page. It should be free from grammatical, spelling and punctuation errors and must be thoroughly edited.

7. **HEADINGS**: All the headings should be in a 10 point Calibri Font. These must be bold-faced, aligned left and fully capitalised. Leave a blank line before each heading.

8. **SUB-HEADINGS**: All the sub-headings should be in a 8 point Calibri Font. These must be bold-faced, aligned left and fully capitalised.

9. **MAIN TEXT**: The main text should follow the following sequence:

   INTRODUCTION

   REVIEW OF LITERATURE

   NEED/IMPORTANCE OF THE STUDY

   STATEMENT OF THE PROBLEM

   OBJECTIVES

   HYPOTHESES

   RESEARCH METHODOLOGY

   RESULTS & DISCUSSION

   FINDINGS

   RECOMMENDATIONS/SUGGESTIONS

   CONCLUSIONS

   SCOPE FOR FURTHER RESEARCH

   ACKNOWLEDGMENTS

   REFERENCES

   APPENDIX/ANNEXURE

   It should be in a 8 point Calibri Font, single spaced and justified. The manuscript should preferably not exceed ***5000 WORDS***.

10. **FIGURES &TABLES**: These should be simple, crystal clear, centered, separately numbered & self explained, and **titles must be above the table/figure**. **Sources of data should be mentioned below the table/figure**. It should be ensured that the tables/figures are referred to from the main text.

11. **EQUATIONS**:These should be consecutively numbered in parentheses, horizontally centered with equation number placed at the right.

12. **REFERENCES**: The list of all references should be alphabetically arranged. The author (s) should mention only the actually utilised references in the preparation of manuscript and they are supposed to follow **Harvard Style of Referencing**. The author (s) are supposed to follow the references as per the following:

- All works cited in the text (including sources for tables and figures) should be listed alphabetically.
- Use (**ed.**) for one editor, and (**ed.s)** for multiple editors.
- When listing two or more works by one author, use --- (20xx), such as after Kohl (1997), use --- (2001), etc, in chronologically ascending order.
- Indicate (opening and closing) page numbers for articles in journals and for chapters in books.
- The title of books and journals should be in italics. Double quotation marks are used for titles of journal articles, book chapters, dissertations, reports, working papers, unpublished material, etc.
- For titles in a language other than English, provide an English translation in parentheses.
- The location of endnotes within the text should be indicated by superscript numbers.

PLEASE USE THE FOLLOWING FOR STYLE AND PUNCTUATION IN REFERENCES:

BOOKS

- Bowersox, Donald J., Closs, David J., (1996), "Logistical Management." Tata McGraw, Hill, New Delhi.
- Hunker, H.L. and A.J. Wright (1963), "Factors of Industrial Location in Ohio" Ohio State University, Nigeria.

CONTRIBUTIONS TO BOOKS

- Sharma T., Kwatra, G. (2008) Effectiveness of Social Advertising: A Study of Selected Campaigns, Corporate Social Responsibility, Edited by David Crowther & Nicholas Capaldi, Ashgate Research Companion to Corporate Social Responsibility, Chapter 15, pp 287-303.

JOURNAL AND OTHER ARTICLES

- Schemenner, R.W., Huber, J.C. and Cook, R.L. (1987), "Geographic Differences and the Location of New Manufacturing Facilities," Journal of Urban Economics, Vol. 21, No. 1, pp. 83-104.

CONFERENCE PAPERS

- Garg, Sambhav (2011): "Business Ethics" Paper presented at the Annual International Conference for the All India Management Association, New Delhi, India, 19–22 June.

UNPUBLISHED DISSERTATIONS AND THESES

- Kumar S. (2011): "Customer Value: A Comparative Study of Rural and Urban Customers," Thesis, Kurukshetra University, Kurukshetra.

ONLINE RESOURCES

- Always indicate the date that the source was accessed, as online resources are frequently updated or removed.

WEBSITES

- Garg, Bhavet (2011): Towards a New Natural Gas Policy, Political Weekly, Viewed on January 01, 2012 http://epw.in/user/viewabstract.jsp

AN APPROACH TOWARDS EFFICIENT PREFERNCED DATA RETRIVAL BY PRESORTING

SUNITA DWIVEDI
SR. ASST. PROFESSOR
MAKHANLAL CHATURVEDI NATIONAL UNIVERSITY OF JOURANALISM
BHOPAL

DR. ANIL RAJPUT
PROFESSOR
DEPARTMENT OF MATHEMATICS & COMPUTER SCIENCE
CHANDRA SHEKHAR AZAD GOVERNMENT P.G. NODAL COLLEGE
SEHORE

ABSTRACT

Skyline query processing has become an important issue in database research. The popularity of the skyline operator is mainly due to its applicability for decision making applications. In such applications, the database tuples are represented as a set of multidimensional data points and the skyline set contains those points that are the best trade-offs between the different dimensions. Evaluating skylines, existing techniques primarily focus on skyline algorithms to produce skyline after complete scan of dataset where each tuple is compared with every tuple for Skyline selection. This techniques faces serious limitations, namely (1) Decision making lasts after complete dataset scan, and (that is why) (2) Skyline can only be computed after execution of other relational operator. Developed method PRESORT TO GET SKYLINE (PGS) introduced as an extension to existing Block Nested Loop (BNL) method by improve response time in answering skyline. We claim once a tuple is included in skyline candidate list no subsequent tuple can dominate it hence is not required to compare with tuples following it.

KEYWORDS

Skyline Query, Multidimensional Data, Relational Operator, Decision Making Applications.

INTRODUCTION

Skyline queries are a specific, yet relevant, example of preference queries and have been recognized as a useful and practical way to make database systems more flexible in supporting user requirements. This has motivated the development of many algorithms for the efficient skyline evaluation on large Datasets.

Skyline operator, returns a set of tuples not dominated by any other tuples, has widely been applied in preference queries in relational databases. Skyline has a history before introduced in database, as maximal vector or pareto operator [2,8,9]. Its implementation enables maximal vector applications to be built on relational database systems efficiently [6,7].

Given a set of hotels with the attributes of price (price) and distance to the beach (distance). Hotel A dominates hotel B if A.price ≤ B.price, A.distance ≤ B.distance, and strictly A.price < B.price or A.distance < B.distance. The most interesting hotels are called skyline hotels which are not better by any other hotels in price and distance.

It is proposed to build the Skyline operator on top of the other operator as Join and Order By in [3] in relational database. If we start the Skyline evaluation above all operations, then the total cost of the query is the sum of the query cost and Skyline computation. Additionally, this sequential execution causes the Skyline evaluation to be blocked by the other operators. We observe that efficient Skyline evaluation with other relational operator require a continuous execution of both Skyline and the other operator.

In this paper, our goal is to develop an algorithm, to behave efficient execution of skyline query in relational settings. We are aiming to develop an extension to basic BNL produced in [3], BNL requires scanning of the whole dataset to output the first Skyline object which blocks the result up to end of execution. Developed method Presort to Get Skyline (PGS) produced skyline results in pipelined fashion so fusion of skyline with other operators can be further performed.

Similar to BNL, our proposed method PGS also maintains a window and adopts multi-pass processing but has less bookkeeping overhead than BNL. We developed PGS with the idea, when a point (tuple) is added to the window it is already known that the point is non dominated as per preference. The window's points are used to eliminate stream points. Any stream point not eliminated is itself added to the window.

We arrange the tuples in an order by providing a score based on preferences. The tuples with a lower score have higher possibility to dominate a larger number of tuples. It is impossible for a point off the stream to dominate any of the points in the window. Any point is thus known to be maximal at the time it is placed in the window. So if lower score tuples are inserted to the candidates list first, the number of comparisons can be largely reduced that makes the filtering process much quicker. In addition, when a tuple is added to the candidate list of the window, it is for sure a Skyline point since this tuple cannot be dominated by any tuple that has not yet been presented. Such a scored list approach also enables the progressive output of skylines while BNL has to wait till all tuples are being compared. In regards to the window operation developed method does not replace tuples already in window.

RELATED WORK

In application of decision making user preferences consideration is important. Two areas are emerged for implementing preferences. One is using preference functions [1,4]. Preferences are defined by applying a score to each tuple based on attribute values. Many unrealistic preferences are than generated.

Other approach which attracted interest of database research community is based on Skyline Operator. The skyline operator is an elegant summary method over multidimensional datasets. Skyline queries are applied as popular and powerful paradigm for extracting interesting objects from a multidimensional dataset. Given a set r of d-dimensional objects (or points), the skyline of r is the set of nondominated, points in r where domination is defined as a point p dominates point pi if p is at least as good as pi on all the attributes of interest, and strictly better than pi on at least one attribute. SQL syntax for the skyline operator first introduced in [3];

The Skyline operator encapsulates the implementation of the Skyline of clause. Much of the recent Skyline work has focused on designing good algorithms [3,10] that are well-behaved in the context of a relational query engine. Nearest neighbor is closest database problem to skyline evaluation. The implementation uses both indexed and non indexed approaches. Using indexes has its own disadvantages of maintaining indexes.

The basic Block Nested Loop (BNL) introduced in [3] is straight forward way to compute the skyline. In BNL each data point P is compared with all the other data points and report P as a Skyline point if P is not dominated. In implementation of BNL a window of candidate (Skyline) is maintain in main memory and each data element in data set is compared with every candidate element one by one. When a data point P is evaluated, P is compared against all the candidate points in the window. There are three cases:

Case 1 P is dominated by a candidate point in the window, it is discarded immediately without further comparison with other candidate points.

Case 2 P dominates some candidate points in the window, the dominated candidates points are removed from the window and P is inserted into the window as a new candidate point.

Case 3 P is neither dominated by nor dominates any candidate point in the window; P is inserted into the window as a new candidate point.

After all the data points are processed, the candidate points in the window are reported as the skyline. When the window grows larger than the main memory, BNL adopts a temporary file to store the candidate points. Specifically, when the window is full the new candidate point found in Case is written to the temporary file.

After the dataset is scanned, all the candidate points in the window which are processed before the creation of the temporary file are output as part of the skyline. Then, BNL evaluates all data points in the temporary file in the same way again in next pass until no temporary file is created. Finally, all candidate points in the window are output as part of the skyline.

Example 1 We use an example to illustrate BNL algorithm. Consider preference point is close (near) point. Consider computing Skyline of the given dataset shown Table 1 by BNL with window size 3 (can keep 3 points at maximum).

The dataset is processed in the order of the subscript number of each data point. Initially, P1 is inserted into the window as it is empty. The next data point, P2, is processed. Since P1 is far in both dimensions P2, P1 is removed and P2 is inserted into the window. Similarly, after P3 is processed, only P3 is in the window. Then, P4 is inserted into the window as it is not dominate by any data point in the window (i.e., P3). After that, P5 and P6 are processed in the same way and P6 is inserted into the window. When P7 is processed, it is written to the temporary file as the window is full with data points (P3, P4, and P6). The time stamp of this temporary file is 7. At the end of this iteration, the data points in the window are P3, P8, and P9, and the temporary file contains data points P7 and P10. Then, the data points in the window which is processed before time stamp of the temporary file (i.e., 7) are output as Skyline points. In this case, P3 is output. After that, BNL continues to process the temporary file in the same way till no data point is remained in any temporary file. The Skyline result is (P3, P8, P9, and P10). To speed up the comparisons between the data points in question and the candidate points, the window is organized as self-organizing list. In this list, when a candidate point is found dominating other data points, it is moved to the beginning of the list. Consequently, it is first compared when evaluate the next data point. This variant reduces the number of comparisons as the data points which dominate more others may be compared first.

Only Skyline tuples are added to list and same time pipelined to output, all we need is that we can check subsequent tuples for whether they are dominated by this tuple. For this, we only need the tuple's Skyline attributes. Real data will have many attributes in addition to the attributes we are using as Skyline criteria. Therefore we are not maintaining the actual tuple in the window, only Skyline attributes are projected and kept in window for comparison. The Skyline generation worked only for numerical data so Integer and float values are only taking part in Skyline computation.

PRELIMINARY: The property described in [3], for any monotone scoring function M →R if p ∈ M maximizes that scoring function, then p is in the Skyline. In addition, for every point p in the Skyline, there exists a monotone scoring function such that p maximizes that scoring function. In other words, no matter how you weigh your personal preferences towards price and distance of hotels you will find your favorite hotel in the Skyline and Skyline does not contain any hotels which are nobody's favorite. We use this property for sorting tuples on their scores (with a scoring function).

Again we observe that, any nested sort of Relation R over the Skyline attributes $a_1, a_2 ..., a_k$ (sorting in descending order on each), as in the figure 1.1 is a topological sort with respect to the dominance.

Justification of the above is:

Let us assume domains of a_i are define in reals over 0..1 non inclusive. Let a nested query in figure 1.1 is applied.

As R be a finite table there is an $\varepsilon > 0$ such that

for $i \in 1.....k$ for all $r, t \in R$ if $r(a_i) < t(a_i)$ then $t(a_i) - r(a_i) > \varepsilon$.

Consider a scoring function

$T(r)= \sum_{i=1}^{k} (\varepsilon/k)^{i-1}[a_i]$

If $r(a_1) > t(a_1)$ than $T(r) > T(t)$

i.e. T gives total preference to a_i over all $a_{i+1}.....a_k$ This is equivalent to nested sort.

TABLE 1: SKYLINE OF THE EXAMPLE DATASET USING BNL

Point	X dim	Y dim
P1	10	9
P2	6	8
P3	1	7
P4	3	6
P5	7	6
P6	4	5
P7	8	4
P8	2	3
P9	5	2
P10	9	1

This observation leads to our concept, As a result of nested sort query a current tuple is dominated by one of the tuples preceded it only. No following tuple to it will dominate current tuple. An external sort is a well known component in relational engine which can be called using an ORDER BY clause in SQL adaptation of sort need not require any addition in optimizer.

Figure 1 An order by query

SELECT * FROM R

ORDER BY a_1 desc,..., a_k desc;

Let F be dominance function in relation R with n tuples. F(t) can range from 0 to n-1 i.e. dominance of tuple t under definition of F ranges from none to all tuples. If t is input in first pass for dominance at least F(t) tuples get eliminated. Again F(t) is non additive i.e. window tuples may dominate same tuples in common. If we maximize cumulative F of tuples in window will improve reduction factor [5]. We already implemented it comparing to sorted data.

Again with dominance definition even for non Skyline F may high (any non Skyline point may also dominate some other non skyline). For a non Skyline tuple r there exists at least a t such that $F(t) \succ F(r)$ and all s dominated by r all also dominated by t under F. Every time a tuple replaced by a new tuple in window, F for new tuple is higher than a maximum F over the tuple it is replacing when window is full cumulative F is fixed for next pass, and strategy is to fill the window with high F.i.e. high order in sorted list must appear first. It is accomplished by sorting of tuples in ascending order of F. If a tuples F is known to sort off course F will be monotone. Calculating F for each tuple is costly but we can estimate probability that a given tuple dominate an arbitrary tuple.

Probability of $t[a_i] \succ r[a_i]$ is $t[a_i]$ under the assumption that domain of tuple is normalized between reals 0..1,and data in dataset is uniformly distributed and each data is independent.

Probability $t \succ r$ considering all dimensions is

$\prod_{i=1}^{k} t[ai]$

This probability is monotone and can be computed from each tuple. The definition of monotone function is that addition can be applied only on Skyline attributes. So the monotone function can defined as

$E(t) = \sum_{i=1}^{k} \ln(t(ai) + 1)$ has same order as ordering by probability, and as E is an Entropy we use it as Entropy Scoring for our implementation.

Example 2 Take the same dataset shown in Table1 as example. The corresponding normalized coordinates and entropy value of the data points are listed in Figure Suppose the window size is 3 and now after sorting on the entropy values the processing order of data points is p8, p3, p9, p10, p4, p6, p7, p5, p2, and p1. Clearly, the first three points p8, p3, and p9 do not dominate each other. So, all of them are inserted into the window and the window is full now. Then, p10 is processed and it is written to the temporary file as it is not dominated by any data points in the window. After that, all other data points are processed and discarded due to each of them is dominated by one of p8, p3, and p9. Now, the first iteration is done. In the beginning of the next iteration, all the data points in the window are output as the Skyline points as they are processed before the temporary file. Since p10 is the only data point in the temporary file, the process terminates after it.

TABLE 2: CALCULATING ENTROPY ON EXAMPLE DATA

ID	Coordinate	Normalized Coordinate	Entropy Value
P1	10,9	1.0.0.9	1.4
P2	6,8	0.6,0.8	1.06
P3	1,7	0.1,0.7	0.6
P4	,6	0.,0.6	0.7
P5	7,6	0.7,0.6	1.00
P6	4,5	0.4,0.5	0.74
P7	8,4	0.8,0.4	0.92
P8	2,	0.2,0.	0.44
P9	5,2	0.5,0.2	0.59
P10	9,1	0.9,0.1	0.74

PRESORT TO GET SKYLINE (PGS)

As studied in previous section BNL produce results only after a complete cycle of scanning. We attempt to improve BNL with early Skyline result generation. We developed here a new algorithm with the objective of early results on domination test, so the Skyline points are continuously produced and may further combined with other database operation.

In the proposed PGS execution sorting is applied first on input set; Candidate points are inserted into the list in sorted order next and sorted stream is filtered for the non dominance as in BNL. A window is than created as in BNL to hold the Skyline tuples.

The Skyline objects are output to a window. The window is a data structure used to hold the Skyline tuples. We applied a doubly list for implementation of window. The sorted tuples are sweep and checked against tuples in window; objects put into the window are guaranteed to be in the Skyline If current tuple is dominated by any of the window tuple can easily be discarded otherwise it is a Skyline tuple. The current tuple if not discarded can be added as an Skyline and also pipelined to output stream. Each object is compared only with the tuples existing in window not with all the tuples preceded is due to the transitive property [5] of Skyline dominance relation.

If the window is full and cannot accommodate the current tuple can be move to another temporary file. Next comparison is takes place with the elements of window and again either discarded or keeps in window or temporary file. At the end if all the tuples are compared and no temporary file exists we have completed with our results else the algorithm repeats with opening the temporary file as input stream. Multiple passes are only required if window is not large enough to hold all the Skyline tuples.

THE ALGORITHM

Input: (a) Data set D with n tuples.(List of data)
(b)An external sort method (Quick sort is used)
Output: (a) set of Skylinepoints

```
I presot = sort(I); //Presorting dataset
Set O = Φ; W = Φ; Temp = Φ//Initialization
while ¬EOF(I presot )//Loop
p = next(I presot )
isSkyline = true
for each q ∈ W
if q ≻ p //Dominance test in window
free(p);
isSkyline= false;
Break;
elseif p ≻ q //dominated p.
Release(W,q);
If isskyline
If hasfreespace(W)
Add(W,p);// include p in window.
Else
Append(T,p) // write p to temporary file for second pass.
Free(p)//remove p from dataset.
Append(O,p)
EOF(Ipresot)
Ipresot ← Temp ; Temp ← Φ ;
W ← Φ;
Return (O);
```

Experiment and RESULTS: With experiment we provide the features of our developed method and compare with basic BNL on early skyline result generation criteria.PGS produces early results as soon as catches as skyline candidate. Skyline algorithm execution. Our experiment is based on following features.

- Our experiments are carried out on an Intel Pentium IV 2.2 GHz duo CPU machine with 2GB main memory and a 80 GB hard disk, running Microsoft Windows XP.
- We implemented the algorithm on a random dataset on student marks using C++ with 3, 5,7 dimension of Skyline query with 419 Skyline points in d, 876 Skyline points in 5d and 9816 Skyline in 7d comparisons.
- We consider sort phase and Skyline phase as two different operations. The time required to sort the data is included in the execution time shown.
- We execute the methods on 50,000 tuple dataset. A tuple is of 100 bytes with 10 integer values (40 byte) and a 60 byte string value.
- Both behave similar in and 5 dimensions while in 7 dimension the execution time differ, PGS perform better to BNL.
- We observe when BNL is given more main-memory allocation and thus, its window is larger—its performance deteriorates.

FIGURE 2: EXECUTION TIME VS. WINDOW SIZE IN BNL AND PGS

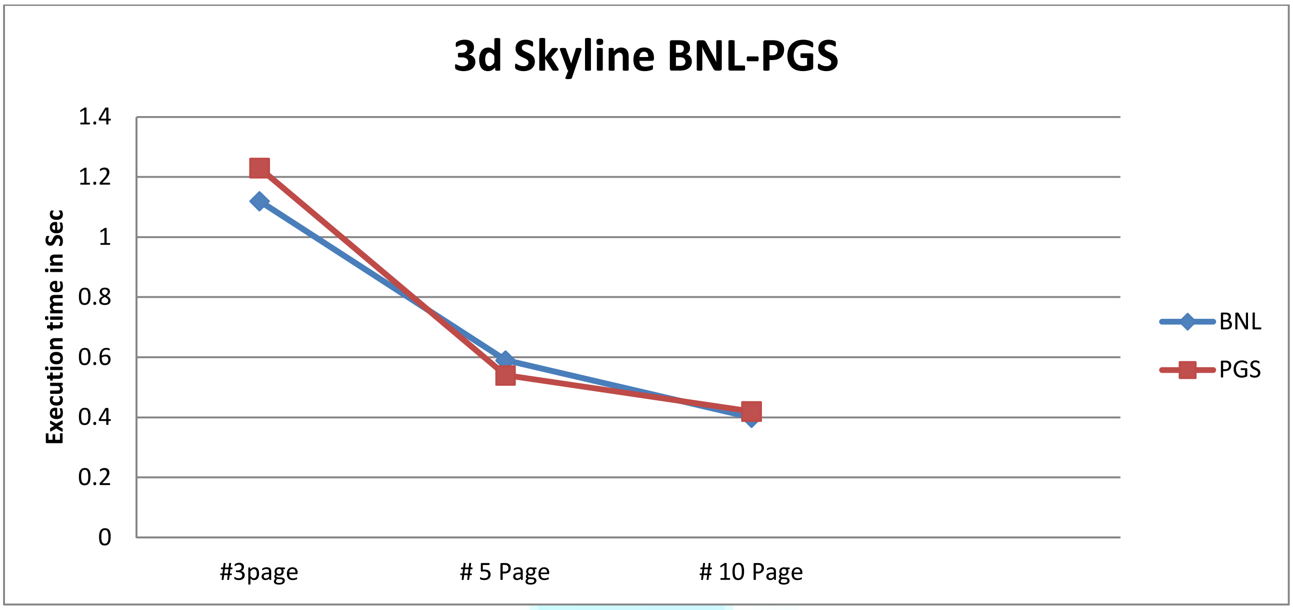


FIGURE 3: COMPARISON BNL AND PGS

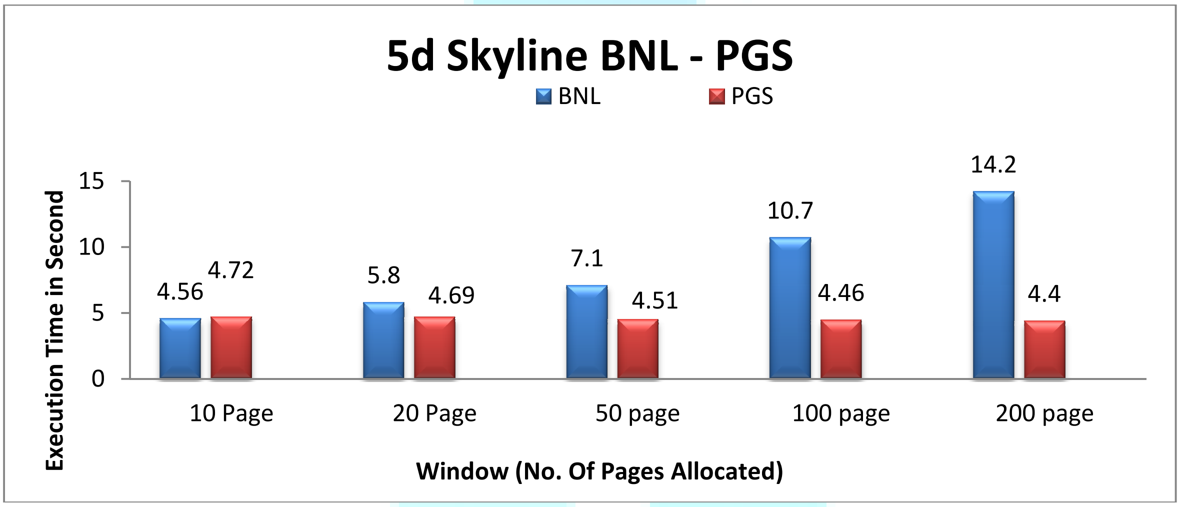
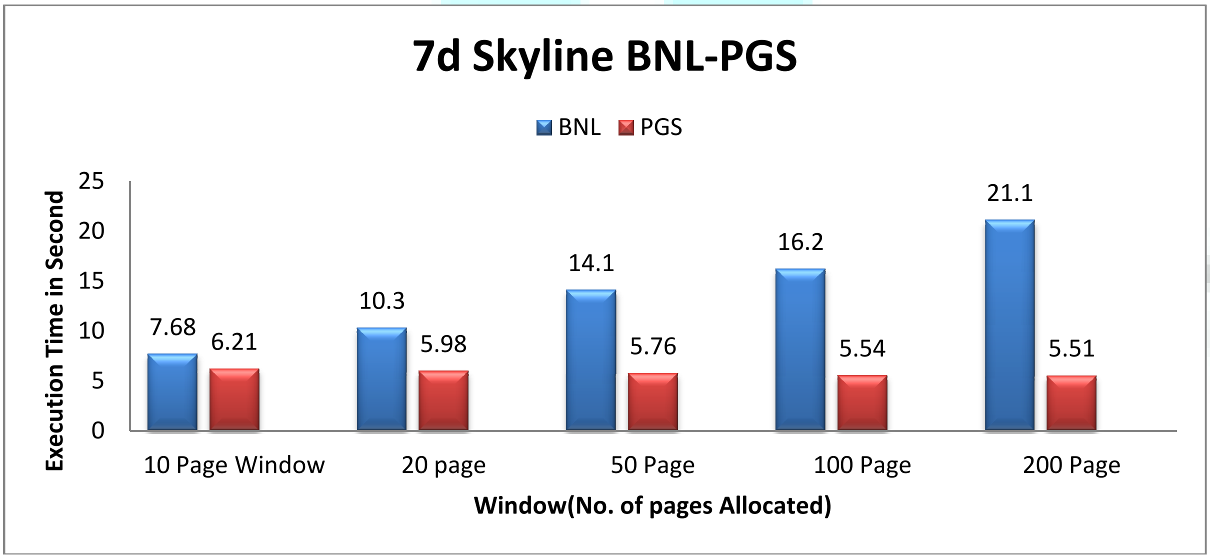


FIGURE 4: BNL-PGS EFFECT OF MEMORY ALLOCATION



CONCLUSION

Skyline operator offers a good start to provide preference query results in relational database. There may be ways to reduce dominance check. We implemented as improvements on BNL. BNL differ with our developed method as BNL is comparing each tuple with other tuple using replacement policy while developed method operate on sorted data set. In PGS management of window is largely simplifies and points in the skyline can be progressively returned without having to wait for all the input to be read

We believe with performance test that our algorithm is a good start to incorporate Skyline computation within other relational operators.

REFERENCES

1. Agrawal R. and Wimmers E. L.. ***"A framework for expressing and combining preferences.",*** *SIGMOD*, pages 297–06, 2000.
2. Bentley J. L., Clarkson K.1., and Levine D. B., ***"Fast linear expected-time alogrithms for computing maxima and convex hulls.",*** Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 179-187, 1990.
3. Borzsonyi S.,Kossmann D. and Stocker K., ***"The skyline operator."*** Proceedings of the 17th International Conference on Data Engineering (ICDE), pages 421-40, 2001.
4. Hristidis V Koudas N., and . Papakonstantinou Y. PREFER, ***"A system for the efficient execution of multi-parametric ranked queries."*** *SIGMOD*, pages 259–270, 2001.
5. Khalefa M., Mokbel M. and Levandoski J., ***"Skyline Query Processing for Incomplete Data.",*** Proceedings of International Conference on Data Engineering (ICDE), pp. 556- 565 ,2008.
6. Kiessling W. ***"Foundations of preferences in database systems.",*** Proceedings of the 28th International Conference on Very Large Data Bases (VLDB). Hong Kong, China. MorganKaufmann, San Francisco, CA, pp.11–22,2002\5
7. Kiessling W. and Koestler G., ***"Preference SQL: Design, implementation, experiences.",*** In VLDB, Aug. 2002.
8. Kung H. T., Luccio F., and Preparata F. P., ***"On finding the maxima of a set of vectors."*** *Journal of the ACM (JACM),* 22(4):pp. 469-476, 1975.
9. Preparata F. P. and M. I. Shamos. ***"Computational Geometry: An Introduction."*** Springer-Verlag, New York, Berlin, etc., 1985.
10. Tan K. L., Eng P. K., and Ooi B. C., ***"Efficient progressive skyline computation.",*** Proceedings of 27th International Conference on Very Large Data Bases (VLDB), pp.01-10, 2001.

REQUEST FOR FEEDBACK

Dear Readers

At the very outset, International Journal of Research in Commerce, IT & Management (IJRCM) acknowledges & appreciates your efforts in showing interest in our present issue under your kind perusal.

I would like to request you tosupply your critical comments and suggestions about the material published in this issue as well as on the journal as a whole, on our E-mail**infoijrcm@gmail.com** for further improvements in the interest of research.

If youhave any queries please feel free to contact us on our E-mail **infoijrcm@gmail.com**.

I am sure that your feedback and deliberations would make future issues better – a result of our joint effort.

Looking forward an appropriate consideration.

With sincere regards

Thanking you profoundly

Academically yours

Sd/-
Co-ordinator

DISCLAIMER

ABOUT THE JOURNAL

In this age of Commerce, Economics, Computer, I.T. & Management and cut throat competition, a group of intellectuals felt the need to have some platform, where young and budding managers and academicians could express their views and discuss the problems among their peers. This journal was conceived with this noble intention in view. This journal has been introduced to give an opportunity for expressing refined and innovative ideas in this field. It is our humble endeavour to provide a springboard to the upcoming specialists and give a chance to know about the latest in the sphere of research and knowledge. We have taken a small step and we hope that with the active co-operation of like-minded scholars, we shall be able to serve the society with our humble efforts.

Our Other Journals