International Journal of Computer Engineering Science (IJCES)
Volume 2 Issue 5 (May 2012) ISSN : 2250:3439
https://sites.google.com/site/ijcesjournal http://www.ijces.com/

Future Research Directions in Skyline Computation

Ms. R.D. Kulkarni[1], Prof. Dr. B.F. Momin[2]

[1] Ms. R.D. Kulkarni, Department of Computer Engineering, K.K.W.I.E.E.R, Nasik, Maharashatra, India,
rupaliwaje@rediffmail.com
[2] *Dr*. B.F. Momin, Department of Computer Science and Engineering, W.C.E. Sangli, Maharashatra, India,
bfmomin@yahoo.com

Abstract: The purpose of this paper is to put focus on the current trends in the area of skyline computation which is a computational trend related to database research community and also to highlight the further research directions in the same. In recent years, the skyline queries or skyline computations have been used in many advanced applications such as location based queries originated from the mobile phones. Although the concept of skyline computation originated from centralized database environment, as the application needs grew, the concept has been applied successfully to modern computational environments like distributed networks, real time systems, mobile ad hoc networks etc. This paper aims at unfolding the various recent methodologies used till now, for the skyline computation and then highlights the future research directions in this area.

Keywords: skyline computation, skyline queries, modern computational environments

1 Introduction

In computational geometry community, a problem known as 'the maximum vector problem' pre-exists prior to the introduction of skyline queries into database research. In recent years, skyline query processing or skyline computation has become an important issue in database research. The popularity of the skyline operator mainly arises due to its applicability in decision making applications. For example, suppose a person is going on holiday to Goa and she is looking for a hotel that is both cheap and close to the beach. Unfortunately, these two goals are complementary as the hotels nearer to the beach tend to be more expensive. However the person is interested in all such hotels that are not worse than any other hotel in both the dimensions. This set of interesting hotels is the "Skyline". From the skyline, one can make the final decision, thereby weighing personal preferences for price and distance to the beach. So the skyline computation is known as the maximum vector problem. This means for every point 'p' in the skyline there exists a monotone scoring function such that 'p' maximizes the scoring function.

For better understanding, consider the following example, explained with respect to figure 1 shown below. In this example, a person driving a car is trying to locate a hotel in Goa, which is both closest from the beach and cheapest in the price. The records show that hotels h1 and h4 dominate hotels h2 and h3 on price and distance dimensions respectively. Thus the skyline computation uses the notion of

"dominance", where a point is said to dominate other point, if it is if it is as good or better in all dimensions and better in at least one dimension.
The skyline queries may involve more than two dimensions. For example, a person may want additional criteria in the hotel search such as good service, good quality of food and good rating. The parameters like database dimensionality, pre-processing of the dataset, geographic vicinity of database, number of querying criteria used etc. have deep impacts on the performance of the computed skyline results.

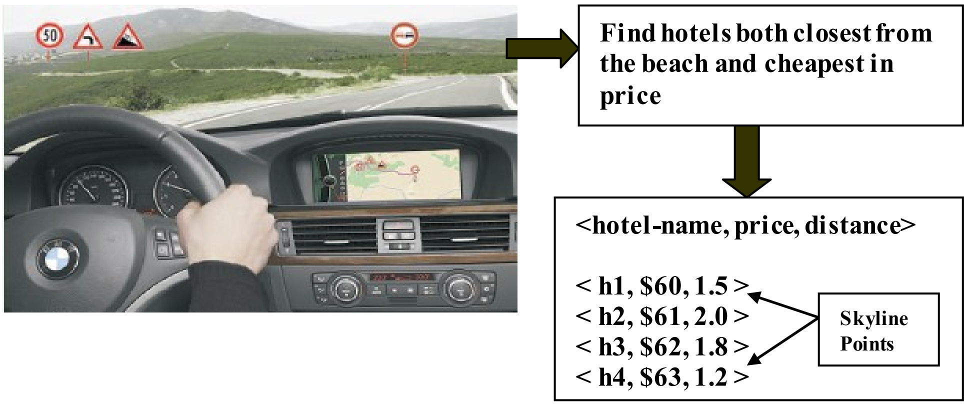

Fig. 1. The Concept of "Dominance" and "Skyline query"

The rest of this paper is organized as follows. Section 2 gives in brief the background and related work done in this area. Section 3 discusses on the future research directions in this area for each of the computational environment and section 4 concludes the paper.

2 Background and Related Work

Skyline computation has been studied in various computational environments like centralized, distributed, real time. It has wide rage of applications in the area of mobile ad hoc networks and web. This section highlights the related work, for each of these computational environments.

2.1 Skyline Computation In Centralized Environment

The concept of skyline queries was put forward for the first time using the notion of the skyline operator [1] by Borzsonyi in 2001. He proposed algorithms such as Block Nested Loop (BNL) and Divide-and-Conquer (D&C). Then the concept of pre processing of the dataset like sorting and indexing aroused. A Sort-Filter-Skyline

(SFS) [2] algorithm as a variant of BNL was proposed. In [3], the authors have provided a comprehensive analysis of those aforementioned algorithms without indexing support and have proposed a new hybrid method, named Linear Elimination Sort for Skyline (LESS). Recently, a Sort and Limit Skyline algorithm (SaLSa) [4] has been proposed. This algorithm is an alternation of SFS and LESS, which presorts the input data using a monotone limiting function and guarantees that during the query processing the non fetched data are all dominated by a stop point. . In [5] the authors have proposed two progressive algorithms: Bitmap and Indexing. The former represents points in bit vectors and employs bit-wise operations, while the latter utilizes iMinMax [6] data transformation and B-tree indexing. Other methods include nearest neighbor- NN [7], Branch-and-Bound Skyline (BBS) [8]. In [9], the authors have discussed the skyline computation in scenario where dataset is vertically distributed.

2.2 Skyline Computation In Distributed Environment

A distributed skyline query can be processed by evaluating multiple constrained skyline queries on different servers. A framework, called SkyPlan [10] has been proposed that maps the dependencies between the queries into a graph and generates cost-aware execution plans. The one of the possible ways to deal with geographically scattered data has been studied using a framework called PadDSkyline [11]. The theme of "incomparability" for skyline computation has also been explored. The authors have proposed and compared skyline computation based on "dominance" and "incomparability" through algorithms BSkyTree-S and BSkyTree-P [12]. The progressive skyline computations using DSL [13] and other algorithms [14] have also been proposed for query load balancing. The advent of multi-core processors is making a profound impact on software development. In [15,16] , the authors have modified the basic skyline computation algorithms SFS (Sort Filter Skyline), BBS (Branch and Bound Skyline) and SSkyline (Simple Skyline) to induce the "possible parallelism" in them and have proposed a new algorithm PSkyline (Parallel Skyline). In [17], the authors have proposed an algorithm called as SSP (Search Space Partitioning) which exploits features of BATON -Balanced Tree Overlay network (also used in [18]) for indexing the dataset so that in structured peer to peer network, the peers will be accessed to the minimum and exact data sub space to compute the skyline can be searched efficiently. Other aspects of parallel skyline computation like rank-aware queries, constrained skyline queries and progressive skyline computation in P2P networks have been proposed in [19-21] respectively. Proper data sub space partitioning is another aspect of parallel skyline computation. The related approaches have been discussed in [22-28].

2.3 Skyline Computation In Real Time Environment

The area of real time skyline computation has raised various research challenges. In real time environment, dataset may tend to be uncertain. Related work can be found in

[29-32]. The work of "probabilistic skyline computation" called PSkyline [29] has been elaborated which uses a new in-memory tree structure called Z-tree. Online retrieval of skyline queries is another aspect in real time processing. The related work can be found in [33-37]. In a number of emerging streaming applications, the data values that are produced have an associated time interval for which they are valid. A useful computation over such streaming data sets is, to produce a continuous and valid skyline summary. An algorithm LookOut [38] continuously evaluates a skyline over multidimensional data in which each element is valid for a particular time range. An algorithm named CSC (Constrained Skyline Computing) [37] incrementally maintains all the non-redundant dominance relationships of tuples over a sliding window and then computes all the constrained skyline points. The k-dominant skyline objects are difficult to maintain if the database is updated. Related algorithm for maintaining k-dominant skyline objects has been discussed in [39]. Currently, only the insert and delete operations are considered by the authors. Yet, update operations have to be considered.

2.4 Skyline Computation in Mobile Environment

Cost of 'communication' and 'query execution' are major concerns in computing when it deals with limited-memory and storage constraints of mobile devices. Another aspect of skyline computation is the one in which skyline query and the dataset on which this query will operate both are continuously changing in spatio-temporal dimensions [40]. When query and data points move over time, re-computing skylines for every change incurs prohibitive cost, so to exploit the spatio-temporal coherence of the movement to predict the changes and minimize the re-computation only to points affecting the skyline changes, seems crucial. In [41] the authors have proposed the method to reduce communication cost among the mobile devices and a method to reduce execution time on each participating mobile device. Another approach of Ring-Skyline (RS) [42] has been proposed which divides the monitoring area into several rings and adopts in-network query processing. Due to privacy concerns and limited precision of localization devices, the input of a user location is often a spatial range. So recently, [43] has proposed I-SKY (index-based) and N-SKY (non-index based) approaches to process the range based queries in mobile environment. Range to Range Skyline Query (R2R) and Point to Range Skyline Query (P2R) are introduced in [44]. Idea of a scoring function, which operates on skyline frequency, ranks the local skyline points that have a high probability of belonging to final skyline set. The spatial skyline queries for location dependent systems have been discussed in [45-47].

2.5 Skyline Computation for the Web

With the proliferation of the Cloud Computing and Software as a Service (SaaS) concepts, more and more web services will be offered, making it difficult for the service providers to choose a web service weighing the various QoS parameters. Related work can be found in [48,49,50]. In [20] the authors have proposed the usage of the concept of skyline queries to select the optimum 'skyline-service' on various QoS parameters. Because number of such services in a web service class, tend to be very huge, they have also proposed a method of selecting "representative-skyline service" combining the concept of skyline with "k-means Clustering technique. CFS-Collaborative Filtering Skyline is [51], a general framework that combines the advantages of CF (Collaborative Filtering) with those of the skyline operator. CFS generates a personalized skyline for each user based on scores of other users with similar behavior. Service composition is emerging as an effective vehicle for integrating existing Web services to create value-added and personalized composite services. Related work exists [52-54] to exploit the dominance relationship among services to find a set of "best" possible composite services, referred to as a composite service skyline. The XML querying support for skyline has been discussed in [55]. The suggested query structures include both value-based and structural-based desires of users. The skyline querying work also exists for clouds. A SkyRank [56] algorithm improves the usefulness of skyline analysis and provides more opportunities for users to analyze and explore the whole data. The usefulness of skyline approach has been discussed for SOA in [57].

Thus a lot of research has been done in the area of skyline computation related to the various computing environments. However with emerging trends in communication technologies, database technologies, parallel processing architectures, web technologies, the applications based on skyline computations do find a very strong research scope. In the next section, the future research directions in this area have been discussed for each of the above computational environment.

3 Research Directions In Future

Because of the emerging technologies in database community, lots of research efforts are needed to map the skyline computation requirements on to these emerging technologies. In this section, such possible research directions have been discussed, which seem obvious after doing the literature survey of skyline computation methodologies on above various computational environments.

3.1 Centralized Environment

This was the computational environment, where from the research on skyline computation began. After the basic algorithms line BNL, D&C proposed by Borzsonyi in 2001, various techniques to improve efficiency of the computation were developed like pre processing of the dataset using sorting, indexing. However, the work done till now, have not considered the following points.

Effect of the indexing structures on skyline computation: Indexing structures for the database like B tress, B+ trees, R trees, BATON etc. improve the accessibility of the data records to be used for the computation. A comparative study is required to

determine which indexing structure is useful when various parameters of the computation like database cardinality, number of querying dimensions, communication cost, frequency of the repeat access of the data records etc. are put on the test. If this estimation of usage of the proper indexing structure, for a given multi criteria computation can be made available, it will be an obvious reason for great improvement in efficiency of the skyline computation.
Effect of the database qualities on skyline computation: What sort of database qualities and skyline query's qualities yield the most efficient computation? The question is difficult to answer unless research is done on the ways, the database qualities do affect the efficiency of the skyline computation. Consider the following scenarios for example. The skyline computation efforts on a database is a waste if the database accessed for the computation is not reliable (not trustworthy), is not available for the time, does not yield quality data, does not produce required clarity. So the before the computation begins, the accurate estimation of reliability, availability, usability, clarity of the database is absolutely necessary. The proper ranking method needs to be developed, which can commence the computation only if quality parameters are in allowed threshold range.
Effect of high voluminous database on skyline computation: In relevance with very large databases, the dataset pre processing techniques like sorting or indexing do not yield efficiency for the computation. Some advanced techniques like 'choosing representative skyline from clustered datasets' can be thought of. If a single database itself is organized in a clustered format, then proper cluster can be accessed for determination of the representative skyline. Applicability of such a technique seems must for a high voluminous database.

3.2 Distributed Environment

Because now a days, data tend to be stored in a distributed manner, need for skyline computation in a distributed environment became significant. Many skyline computation techniques have evolved which for the sake of computation, focus on parameters like, proper region partitioning methodologies to take care of geographic or spatial qualities of data servers (minimization of the communication cost), methodologies to achieve independent, parallel computation in each region (maximization of the parallel computation), application of indexing structures like BATON (maximization of the accessibility), techniques to improve I/O cost (like caching, data streaming etc). However following points still find scope for the further research.

1. Effect of parallelizing the computations: Once proper regions of data servers have been formed, a novel method such as estimation of the candidate data server for parallelization of the computation can be implied. Giving more details, suppose a particular data server holds data which is small in volume as compared to other data servers. For this server, any normal, pre defined skyline computation technique can be applied. However, if a particular data server holds a high voluminous dataset, then there exists a scope of parallelizing the skyline computation. This means sequential algorithm can

be re devised so as to employ possible parallelism in the code. For small sized data server, cost of converting sequential algorithm into parallel one or applying the pre decided parallel algorithm, may not yield intended efficiency. So a technique which gives estimation of the candidate data portion or data sets for applying parallel computations and then applies it on them , seems very much intended.

2. Effect of network failures on skyline computation: To the best of our knowledge, there is no explicit focus given on this aspect when it is obvious that in distributed environment, the node or network failures will be obvious risk carrying scenarios for getting the final results. When region partitioning algorithm works, it should have built in capability to estimate proper data server as a candidate to hold the intermediate skyline computation results. To help achieve this task, may be explicit indexing structure to point to data servers which hold the possible intermediate results, can be devised. Another alternate solution can comprise utilization of cache results for recovery process. It is obvious that, these techniques do not take any resemblance with the database recovery techniques of distributed databases. The reason being, the proposed techniques are supposed to have a stronger skyline computation algorithm itself, which is self sufficient for result recovery process. Here utilization of any database management system's tools is not done, in order to relieve the DBMS to exists with its original configurations.

3.3 Mobile Environment

The typical application of skyline computation in mobile ad hoc networks, include location based multi preference queries. In this computing environment, the most complex scenario is that, both the query generating source and the datasets on which query is fired, are moving. While the skyline computation is in progress, the most major focus, related to this environment, is at reducing communication and query execution costs. In these concerns, following research goals seem significant.

1. **Effect of data region overlapping:** Mobile devices have inherent limitations of limited processing memory. Hence to minimize the communication and query execution costs, the participating datasets must be partitioned in such a way that no duplicity of processing efforts occur when data tend to be duplicate. A strategy can be devised, where each mobile device announces, its representative dataset range and accordingly, the regions for computing the skyline may be then formed. The existing algorithms have not taken this into consideration.
2. **Effect of global VDR values:** The existing skyline computation algorithms, take into consideration, the global VDR (Volume of Dominating Region) values. To decide the local cut offs of the involved computing dimension of the database, at each of the mobile device. These values are assumed to some constant and named as global VDRs, wrong assumptions of which can generate totally misleading results. These global VDRs, should be computed,

considering only the participating mobile devices, instead of borrowing them from previously known values, which may appear false, in the next computing moment. Inculcating this feature will surely generate, better results of the skyline computation.

3.4 Web based Environment

It is increasingly becoming difficult for the service providers to choose a web service weighing the various QoS parameters, when they offer a set of web services through their web sites. The theory behind the skyline computation has been used to guide the web service providers to weigh their personal preferences against the various QoS parameters and then select optimum set of services for improving their business. Consideration of the following points can further make these techniques more efficient.

1. Effect of run time QoS values: In the era of cloud computing and SaaS (Software as a Service), huge number of web services do exist which serve the common purpose. So because of heavy competition in the service developers, the market values of such services tend to be dynamic, changing with the competing scenario! So consideration of the static QoS values for service selection using the skyline computation may lead to a wrong guideline for the service providers. A skyline computation methodology needs to be developed, which can estimate a runtime QoS value for a service and incorporate this value for betterment in service selection process.
2. Privacy preserving skyline computation: Since much of user data, now increasingly put online, users do have serious privacy concerns about the usage of this data. Existing skyline computation techniques, incorporated in LBS (Location Based Service) applications tend to hide the user’s location. However mere location need not be the concern of user’s privacy. Newer skyline computation techniques can be devised which before the skyline computation process, use basic privacy preservation methods like suppression, generalization etc. on sensitive dimensions of the database, thus hiding the sensitive data and still serving the user’s queries.

Thus in this section, we have explored the various future research directions for the skyline computation, on various computational environments. This discussion has clearly highlighted, that the scope for research efforts in this area is still wide.

4 Conclusion

In this paper, we have discussed the possible, future research directions for skyline computation. Many more may evolve, after reading this and the related surveyed papers. It is bound to be true that, with the emerging trends in database technologies, communication technologies, web technologies, data server architectures and growing user's needs, the scope for the research in this area is wide open. Even though, one of the above technologies finds a change, variety of skyline computation techniques will definitely evolve.

References

1. S. Borzsonyi, D. Kossmann, and K. Stocker, "The Skyline Operator," Proc. IEEE Int'l Conf. on Data Engineering, (ICDE), pp. 421-430, 2001.
2. Chomicki, P. Godfrey, J. Gryz, and D. Liang, "Skyline with Presorting," Proc. IEEE Int'l Conf. Data Eng. (ICDE), 2003, pp. 717-719.
3. P. Godfrey, R. Shipley, and J. Gryz, "Maximal Vector Computation in Large Data Sets," Proc. IEEE Int'l Conf. Very Large Data Bases (VLDB), 2005, pp. 229-240.
4. Bartolini, P. Ciaccia, and M. Patella, "Efficient Sort-Based Skyline Evaluation," ACM TODS 2008, vol. 33, no. 4, pp. 1-45.
5. K.-L. Tan, P.-K. Eng, and B.C. Ooi, "Efficient Progressive Skyline Computation," Proc. IEEE Int'l Conf. Very Large Data Bases (VLDB),2001, pp. 301-310.
6. B.C. Ooi, K.-L. Tan, C. Yu, and S. Bressan, "Indexing the Edge: A Simple and Yet Efficient Approach to High-Dimensional Indexing," Proc. ACM Symp. on Principle Database Systems (PODS), 2000, pp. 166-174.
7. D. Kossmann, F. Ramsak, and S. Rost, "Shooting Stars in the Sky: An Online Algorithm for Skyline Queries," Proc. Conf. Very Large Data Bases (VLDB), 2002, pp. 275-286.
8. D. Papadias, Y. Tao, G. Fu, and B. Seeger, "An Optimal and Progressive Algorithm for Skyline Queries," Proc.of ACM Special Interest Group on Management of Data (SIGMOD), 2003, pp. 467-478.
9. George Trimponias, Ilaria Bartolini, Dimitris Papadias, Yin Yang, "Skyline Processing on Distributed Vertical Decompositions", IEEE TKDE, 2011.
10. João B. Rocha-Junior, Akrivi Vlachou, Christos Doulkeridis, and Kjetil Nørvåg, "Efficient Execution Plans for Distributed Skyline Query Processing", Conf on EDBT, March 2011.
11. Lijiang Chen, Bin Cui, Hua Lu, "Constrained Skyline Query Processing against Distributed Data Sites" IEEE TKDE, Vol. No 23 no 2, February 2011, pp. 204-217.
12. Jngwuk Lee, Seung-won Hwang. "BskyTree: Scalable Skyline Computation Using A Balanced Pivot Selection.", EDBT 2010, March 22-26, Lausann, Switzerland, 2010, pp. 195-206.
13. Ping Wu, Caijie Zhang, Ying Feng, Ben Y. Zhao, Divyakant Agrawal,, and Amr El Abbadi "Parallelizing Skyline Queries for Scalable Distribution", EDBT 2006, LNCS 3896, pp. 112–130.
14. Lin Zhu, Shuigeng Zhou, Jihong Guan, "Efficient Skyline Retrieval on Peer-to-Peer Networks", Future Generation Communication and Networking (FGCN), 2007, pp. 309-314.
15. Hyeonseung Im, Jonghyun Park, Sungwoo Park, "Parallel skyline computation on multi-core architectures", IEEE Int'l Conf. on Data Engineering (ICDE), 2009, pp. 808-823.
16. Adan Cosgaya-Lozano, Andrew Rau-Chaplin, Norbert Zeh, "Parallel Computation of Skyline Queries", IEEE HPCS'07, 2007.
17. Shiyuan Wang, Beng chin Ooi, Anthony Tung, Lizhen Xu, "Efficient Skyline Processing on Peer to Peer Networks", IEEE Int'l Conf. Data Eng. (ICDE), 2007.

18. Bin Cui, Lijiang Chen, Linhao Xu, Hua Lu, Guojie Song, and Quanqing Xu, "Efficient Skyline Computation in Structured Peer-to-Peer Systems", IEEE TKDE, Vol. 21, No. 7, July 2009, pp. 1059-1062.
19. K. Hose, M. Karnstedt, A. Koch, K. Sattler, and D. Zinn, "Processing rank-aware queries in P2P systems". In Proceedings of DBISP2P, 205, pp. 238–249.
20. B. Cui, H. Lu, Q. Xu, L. Chen, Y. Dai, and Y. Zhou, "Parallel Distributed Processing of Constrained Skyline Queries by Filtering," IEEE. Int'l Conf. Data Eng. (ICDE), 2008, pp. 546-555.
21. L. Chen, B. Cui, H. Lu, L. Xu, and Q. Xu, "iSky: Efficient and Progressive Skyline Computing in a Structured P2P Network", Proc.. Int'l Conf. Distributed Computing Systems (ICDCS), 2008, pp. 160-167.
22. Akrivi Vlachou, Christos Doulkeridis, Yannis Kotidis, Michalis Vazirgiannis, "SKYPEER: Efficient Subspace Skyline Computation over Distributed Data",. IEEE Int'l Conf. Data Engineering (ICDE) 2007, pp. 416-425.
23. Yufei Tao, Xiaokui Xiao, Jian Pei, "Efficient Skyline and Top-k Retrieval in Subspaces", IEEE TKDE 2007, pp. 1072-1088.
24. J. Pei, W. Jin, M. Ester, and Y. Tao. "Catching the best views of skyline: A semantic approach based on decisive subspaces", proceedings of VLDB, 2005, pages 253–264.
25. Y. Tao, X. Xiao, and J. Pei, "SUBSKY: Efficient computation of skylines in subspaces", IEEE Int'l Conf. Data Engineering (ICDE) 2006.
26. Akrivi Vlachou, Christos Doulkeridis, Yannis Kotidis, Michalis Vazirgiannis,", Efficient Routing of Subspace Skyline Queries over Highly Distributed Data", IEEE TKDE, Vol. 22, No. 12, December 2010, pp. 1694-1708.
27. George Valkanas and Apostolos N. Papadopoulos, "Efficient and Adaptive Distributed Skyline Computation", Proc. of Int'l Conf. on Scientific and statistical database management,", (SSDBM), 2010.
28. Wen Jin, Anthony K. H. Tung, Martin Ester, Jiawei Han3, "On Efficient Processing of Subspace Skyline Queries on High Dimensional Data", IEEE SSDBM, 2007
29. Dongwon Kim, Hyeonseung Im, and Sungwoo Park, "Computing Exact Skyline Probabilities for Uncertain Databases", IEEE TKDE, 2011.
30. Xiaofeng Ding and Hai Jin, "Efficient and Progressive Algorithms for Distributed Skyline Queries over Uncertain Data", IEEE TKDE, 2011.
31. Dimitris Sacharidis, Anastasios Arvanitis, Timos Sellis, "Probabilistic Contextual Skylines", Proc. IEEE Int'l Conf. Data Eng. (ICDE), 2010, pp. 273-284.
32. Wenjie Zhang, Xuemin Lin, Ying Zhang, Wei Wang, Jeffrey Xu Yu, "Probabilistic Skyline Operator over Sliding Windows", IEEE. Int'l Conf. Data Eng. (ICDE), 2009, pp.1060-1071.
33. Bin Jiang, Jian Pei, "Online Interval Skyline Queries on Time Series", IEEE Int'l Conf. Data Eng. (ICDE), 2009, pp. 1036-1043.
34. Michael Morse Jignesh M. Patel, "Efficient Continuous Skyline Computation", IEEE. Int'l Conf. Data Eng. (ICDE), 2006.
35. Yu-Ling Hsueh, Roger Zimmermann, Wei-Shinn Ku, Yifan Jin, "SkyEngine: Efficient Skyline Search Engine for Continuous Skyline Computations", IEEE Int'l Conf. Data Eng. (ICDE), 2011, pp.1316-1319.
36. Su Min Jang, Jae Soo Yoo, "Processing Continuous Skyline Queries in Road Networks", IEEE Symposium on Computer Science and its Applications, 2008, pp.353-356.
37. Jin-xian Lin, Jing-jing Wei, "Constrained Skyline Computing over Data Streams", IEEE Int'l Conf. on e-Business Engineering, 2008, pp. 155-161.
38. Michael Morse Jignesh M. Patel, "Efficient Continuous Skyline Computation", IEEE. Int'l Conf. Data Eng. (ICDE), 2006.

39. Md. Anisuzzaman Siddique, Yasuhiko Morimoto, "Efficient Maintenance of k-dominant Skyline for Frequently Updated Database", Second International Conference on Advances in Databases, Knowledge, and Data Applications, 2010. pp. 107-110.
40. Mohammad Alrifai, Dimitrios Skoutas, Thomas Risse, "Selecting Skyline Services for QoS based Web Service Composition", Conf. on WWW, April 2010.
41. Zhiyong Huang1, Christian S. Jensen2, Hua Lu1, Beng Chin Ooi1, "Skyline Queries against Mobile Lightweight Devices in MANET", IEEE. Int'l Conf. on Data Engineering (ICDE), 2006.
42. Hailan Shen, Zhigang Chen, Xiaoheng Deng, "Location-based Skyline Queries in Wireless Sensor Networks,". IEEE Int'l Conf. on Networks Security, Wireless Communications and Trusted Computing, 2009, pp. 391-395.
43. Xin Lin, Jianliang Xu, and Haibo Hu, "Range-based Skyline Queries in Mobile Environments", IEEE TKDE, 2011.
44. Zhefeng Qiao, Junzhong Gu, Xin Lin and Jing Chen, "Privacy-Preserving Skyline Queries in LBS",. Conf. on Machine Vision and Human-machine Interface, 2010, pp. 499-504.
45. Reeba Rose S,Kavitha V.R, "An Efficient Location Dependent System Based On Spatial Sky Line Queries", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Vol.1, Issue-ETIC2011, January 2012, pp. 27-29.
46. YingyuanXiao, Kevin Lü, Huafeng Deng, "Location-dependent Skyline Query Processing in Mobile Databases", Seventh Web Information Systems and Applications Conference, 2010, pp. 3-8.
47. Wen Jin, Michael D. Morse, Jignesh M. Patel, Martin Ester, Zengjian Hu, "Evaluating Skylines in the Presence of Equijoins", IEE Int'l Conf. Data Eng. (ICDE), 2010, pp. 249-260.
48. Mohammad Alrifai, Dimitrios Skoutas, Thomas Risse, "Selecting Skyline Services for QoS based Web Service Composition", Conf. on WWW, April 2010.
49. Karim Benouaret, Djamal Benslimane, Allel Hadjali, "On the Use of Fuzzy Dominance for Computing Service Skyline Based on QoS", IEEE Int'l Conf. on Web Services, 2011, pp. 540-547.
50. Limin Pan, Liang Chen,Jian Hui,Jian Wu, "QoS-based distributed service selection in large-scale web services", IEEE Int'l Conf. on Services Computing, 2011 pp. 725-726.
51. Ilaria Bartolini, Zhenjie Zhang, and Dimitris Papadias, "Collaborative Filtering with Personalized Skylines", IEEE TKDE 2009.
52. Qi Yu,Athman Bouguettaya, "Efficient Service Skyline Computation for Composite Service Selection", IEEE TKDE,2011.
53. Qi Yu, Athman Bouguettaya, "Computing Service Skylines over Sets of Services", IEEE Int'l Conf. on Web Services, 2010, pp. 481-488.
54. Karim Benouaret, Djamal Benslimane, Allel Hadjali, Mahmoud Barhamgi," Top-k Web Service Compositions using Fuzzy Dominance Relationship", IEEE Int'l Conf. on Services Computing, 2011. pp. 144-151.
55. Sara Cohen, Maayan Shiloach, "Flexible XML Querying using Skyline Semantics", IEEE Int'l Conf. Data Eng. (ICDE), 2009, pp. 553-564.
56. Zhenhua Huang, Yang Xiang, "Improve the Usefulness of Skyline Analysis in Cloud Computing Environments", Second International Symposium on Computational Intelligence and Design, 2009, pp. 325-328.
57. Zhenhua Huang, Yang Xiang, "An Efficient Method for Diversifying Skylines in SOA Environment", Conf. on Intelligent Computing and Cognitive Informatics, 2010, pp. 466-469.