It's alternating Sorted Access to s1 and s2. This pulling strategy is well-defined by "HRJN" [11] and Fagin's "Combined Algorithm" as regards sorted access. Fagin's "Combined Algorithm" forces to perform all random access. When two services are characterized by dissimilar page sizes, the round robin strategy is adapted to select the service with the least depth, so as to make sure that both services are explored up to same depth.

2) Score-Aware (SA)

Score-Aware strategy [9] that decides which service to access next based on the scores of the retrieved tuples by comparing their bounds. This strategy is used in HRJN*[10]. SA produces better results but takes much more time i.e. Cost is higher in compare with Round Robin.

3) Cost-Aware with Random and Sorted Access (CARS):

CARS is the pulling strategy that defined in previous section. In this particular data access method aim is to minimize the COST at the same time maximizing the Top-k combinations to be found.

When given the input of P_1 23 and P_2 20, CARS along with Rank-Join algorithm is used to generate the results then to find the top 100 hotel restaurant combination in just 15 miliseconds. In comaparison with the results of Round Robin Strategy, Round Robin takes 39 miliseconds.

Rank	HotelName	RestaurantName	Street	City	Score
1	Rama International	Dominos Pizza	Jalna Road	Aurangabad	50
1	Ambassador Ajanta	Dominos Pizza	Jalna Road	Aurangabad	50
2	Rama International	Golden Crown Restaurant	Jalna Road	Aurangabad	40
2	Rama International	Kream N Krunch	Jalna Road	Aurangabad	40
2	Ambassador Ajanta	Golden Crown Restaurant	Jalna Road	Aurangabad	40

Figure 4: Top-k hotel restaurant combination on given street

Result Analysis: The Results obtained on the given datasets are shown in fig 3**.**

As CARS strategy generates way better results than Round Robin method. Now in the following section overlook the individual parameters i.e. page size, top-k combinations found, score distribution and cost.

Page Size – page size is individual & one kind of input parameter. Search services usually return the results in pages, the number of results on each page may be large sometimes. For that reason characterize the search service result with a page size P_i.

Top-k combinations found – Fig. 5 shows the result of CARS method and Round Robin method when required to find top-k hotel restaurants. As given in figure when given the input of P_1=23 and P_2=20, CARS is able to find top-100 combinations where Round Robin is only able to find 20 combinations. This Experiment has tested the result with many different inputs still CARS produces better results in all cases.

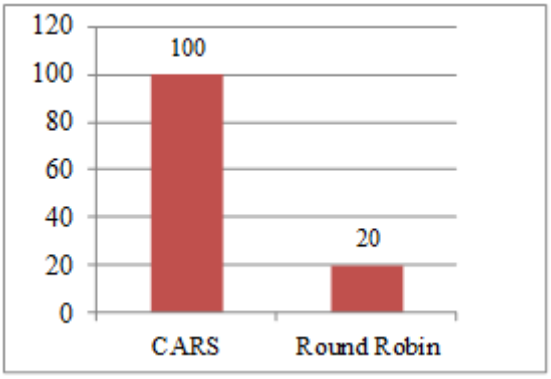

Figure 5: Top-k combination found

Score Distribution – Fig. 6 depicts the results of score distributions of CARS and Round Robin method. In score distribution the maximum score is actually the score of the top most combination and minimum score is score of low most combination. The formula to calculate the score is Hotel star * Restaurant rating. The score distribution of CARS methodology is way better than Round Robin. The maximum score generated by CARS is 50; this is highest score that can generate in any case using any methodology. The minimum score generated by CARS is 21. Where, Round Robin method generates the maximum score of 20 and minimum score of 5. Though given the different inputs CARS is able to produce long range of score distribution. So CARS method has long range to produce the top-k combination while Round Robin, in comparison, has very narrow range to produce the top-k combination.

Cost – Cost is the most important and algorithm defining factor here, upon this cost parameter experiment is able to prove which methodology is better. As in fig. 7 we can observe that, after the execution the Round Robin method takes 39 MS to execute and produce the result. While on the other hand CARS methodology which uses Rank-Join algorithm is able to produce the required top-k combination in 15 MS. So in comparison CARS method is way ahead of Round Robin.

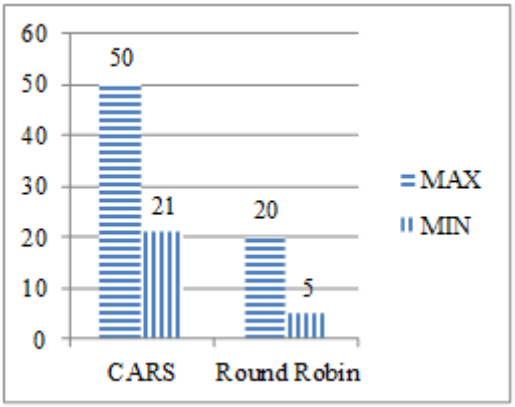

Figure 6: Score Distribution

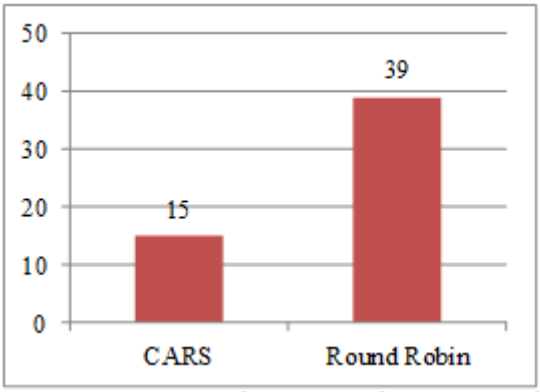

Figure 7: Overall Cost

From the results of CARS methodology and the performance analysis seen in the previous figures CARS Method has produced a cost optimized strategy that uses rank-join algorithm. The performance of CARS method is cost optimized.

6. Related work

Threshold algorithm [8] scans multiple lists, showing different rankings of the same set of objects. An upper bound *T* is maintained for the overall score of unseen objects. The upper bound is computed by applying the scoring function to the partial scores of the last seen objects in different lists. Each newly seen object in one of the lists is looked up in all other lists, and its scores are aggregated using the some scoring function to get the overall score. All objects with total scores that are greater than or equal to *T* can be reported. The algorithm halts after returning the K^{th} output.

The Rank-Join algorithm [1] [8] integrates the ranking and joining tasks in one efficient operator. Rank-Join Algorithm describes the main Rank-Join procedure. The Rank-Join algorithm scans input lists (the joined relations) in the order of their scoring predicates. Join results are discovered incrementally as the algorithm moves down the ranked input relations. For each join result *j*, the algorithm computes a score for *j* using a score aggregation function F. The algorithm maintains a threshold T bounding the scores of join results that are not discovered yet. The top-k join results are obtained when the minimum score of the k join results with the maximum *F*() values is not below the threshold *T*.

7. Conclusion

Encouraged by the goal of answering multi-domain queries with cost optimization propose an execution strategy in this paper, which retrieves top-k combinations that can be formed by joining the results of heterogeneous search services. By using random and sorted access this paper have defined optimized cost aware strategy with an additive cost model.

This paper have successfully implemented Rank-Join algorithm to achieve optimality in terms of cost as well as accuracy by using both access methods i.e. random and sorted data access.

In the future work, the query optimization framework can be extended to the non-additive cost model that will access the services in parallel along with pipelining the joins.

References

[1] Davide Martinenghi, Marco Tagliasacchi, "Cost-Aware Rank Join with Random and Sorted Access," IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 12, DECEMBER 2012.
[2] Search Computing - Challenges and Directions, M. Brambilla and S. Ceri, eds. Springer, Mar. 2010.
[3] Nicolas Bruno, Surajit Chaudhuri, Luis Gravano, "Top-k selection queries over relational databases: Mapping strategies and performance evaluation" ACM Transactions on Database Systems 27(2), 153–187 2002.
[4] Karl Schnaitter, Neoklis Polyzotis, "Evaluating Rank Joins with Optimal Cost," ACM 978-1-60558-108-8/08/06, Vancouver, BC, Canada, 2008.
[5] Christian A. Lang, Yuan-Chi Chang, John R. Smith, "Making the Threshold Algorithm Access Cost Aware" IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 16, NO. 10, OCTOBER 2004.
[6] Jonathan Finger, Neoklis Polyzotis, "Robust and Efficient Algorithms for Rank Join Evaluation" SIGMOD'09, June 29–July 2, 2009, Providence, Rhode Island, USA, ACM 978-1-60558-551-2/09/06, 2009.
[7] Reza Akbarinia, Esther Pacitti, Patrick Valduriez, "Best Position Algorithms for Top-k Queries" ACM 978-1-59593-649-3/07/09, VLDB '07, September 23-28, Vienna, Austria, 2007.
[8] IHAB F. ILYAS, GEORGE BESKALES, and MOHAMED A. SOLIMAN, "A Survey of Top-k Query Processing Techniques in Relational Database Systems", ACM Comput. Surv. 40, 4, Article 11, ACM 0360-0300/2008/10-ART11, October 2008.
[9] J. Finger and N. Polyzotis, "Robust and Efficient Algorithms for Rank Join Evaluation," Proc. 35th ACM SIGMOD Int'l Conf. Management of Data, pp. 415-428, 2009.
[10] K. Schnaitter, J. Spiegel, and N. Polyzotis, "Depth Estimation for Ranking Query Optimization," Proc. ACM SIGMOD Int'l Conf. Management of Data (VLDB), pp. 902-913, 2007.

[11] I.F. Ilyas, W.G. Aref, and A.K. Elmagarmid, "Supporting Top-k Join Queries in Relational Databases," VLDB J., vol. 13, no. 3, pp. 207-221, 2004.

Author Profile

Sudhir G. Chavan has received the B.E. degree in Computer Science and Engineering from Marathwada Institute of Technology, Aurangabad, Maharashtra, India in 2010. During 2011-12 he worked as a lecturer in Maharashtra Institute of Technology, Aurangabad. He is currently pursuing Master of Engineering in Software Engineering with dissertation topic on Cost Optimized Data Access with Rank-Join. His area of interest for research is Database management system, Query processing and Optimization, ranking queries.

Vijay B. Patil has received the B.E. and M.E degree in Computer Science and Engineering in 2004 and 2010 respectively. He has total 10 years of Experience. His research work area is Database Management and Data mining.