Ikebana: Reducing Selectivity Dimensions with Minimial Impact on Plan Bouquet

Adarsh Patil, Pradeep Bansal

November 28, 2014

1 Introduction

In the pretext of the Plan Bouquet style of query execution proposed earlier [Anshuman and Jayant2014] compile time selectivity estimation process was eliminated and replaced by a series of cost limited plan executions at run time to "discover" selectivity of predicates. Statistical selectivity estimation of predicates has been researched for several decades and several techniques have been proposed but all of these suffer from well known drawbacks. Query processing without selectivity estimation was first explored and analyzed in the Bouquet paper and showed theoretical guarantees on sub-optimality. However to make the Bouquet Approach more viable we need to identify the ESS(Error-Prone Selectivity Space) as a first step. The bouquet algorithm performance is predicated and executed on this ESS Space. The number of plans and budgeted costs for execution increase exponentially with the number of dimensions in the ESS. To make bouquet more practical, robust, implementable in OLTP systems and bridge the gap between traditional query processing that is done with selectivity estimation, we need to reduce the number of dimensions along which the bouquet is executed. Also, from real world queries as literature has time and again proven that predicates are not always independent and the AVI property is often violated. For example common scenarios are (i)Selectivities of the form column op column can be accurately predicated with current techniques (ii) The join selectivities of PK-FK joins can be estimated accurately if the entire PK-relation participates in the join. In the process of eschewing selectivity dimension we need to ,also, consider the impact of it on the performance of the bouquet. In this work, we define and derive an expression for maximum sub-optimality of bouquet in the reduced error prone dimensions in comparison with the oracle's query execution cost.Further, we propose a parallelized algorithm titled *Ikebana* to implement single selectivity-dimension reduction given n dimension ESS and show results from TPC-H Q5 having 3D ESS.
The rest of this paper is divided as follows. Section 2 disambiguates and accurately defines the framework of the problem. Section 3 gives an algorithm for single selectivity dimension reduction. Section 4 defines and derives the bounds and analysis of MSO for the reduced bouquet execution. Section 5 shows experimental results on the TPC-H benchmarks queries for dimension reduction. Section 5 concludes the work, draws inference and proposes future enhancements.

2 Problem Framework

2.1 Orthogonal Techniques

In this section we first disambiguate some of the misconceptions of the problem statement, then clearly define the problem and describe the approach we have followed to solve it in rest of the report. The problem is analyzed from the perspective of performance impact of selectivity dimensions.
Given the cost of execution at each point in ESS,one can come up with a function approximation describing cost in terms of ESS as the parameters for the function. This can be done using the well studied algorithms for function approximation. Once such a function has been constructed, we can find the partial differentiation(p.d.) w.r.t. each dimension and the one having the lowest value or falling below some given threshold can be eliminated. Problems using this approach are 1)It is an approximation technique which is no better than earlier statistical techniques for estimating selectivities. 2)Finding/Proving MSO bounds for its impact on bouquet would have been difficult.
Secondly, one can also look at all the predicates and analyse the data types of attributes such as INT, VARCHAR, BLOB, FLOAT and their conjunctives such as AND, OR, BETWEEN, LIKE etc. and analyze how a combination of both affect the selectivity. For example $salary < 10000$ is one of the predicate and we know MAX(Salary) = 10000. But this might be difficult to generalize this approach and does not always yield good results.

2.2 Problem Statement

We now clearly define the concept of redundancy' of a dimension w.r.t. its MSO as follows -
Redundant Dimension : A selectivity dimension d_k ,from an ESS with dimensions set $D = d_1, d_2, ..., d_n$,can be said to be a 'redundant' if the plan bouquet identified from ESS($D - d_k$) as well as from ESs(D),gives similar MSO guarantee when evaluated on ESS(D). Similarly ,redundancy can be defined for a subset of dimensions ($D_k \subset D$) also. With the above definition in mind,the goal is to remove the redundant dimension ,which will be referred as dimension reduction, from the original set of dimensions on ESS.

2.3 Solution Characteristics

Based on the above problem definition and deficiencies of the orthogonal techniques we establish some characteristics for the solution to the dimension reduction problem.

1. The POSP and their budgets that are output by Ikebana should cover every point, i.e. be able to execute every point, in the original non-reduced ESS.

2. We expect that due to increase in the budgets for each of the POSP, the plans will cover more points than they are found to be optimal at leading to overlap of plans regions. Let us illustrate this with an example: Every point in the ESS has a plan and its cost of execution at that point as a tuple (P,C). Consider the point R that has least cost for plan P_2 and a cost C_{R2}

in the reduced POSP. The sequence of cost limited bouquet executions in order of increasing budget are $\{P_1, P_2, ...P_k\}$ and corresponding budgets are $\{B_{P1}, B_{P2}, ..., B_{Pk}\}$. If the budget for P_1 exceeds C_{R2}, plan P_1 will complete execution of point R with a higher sub-optimality than intended. We call this characteristic as *plan overlap*

3. The dimension chosen to be reduced should have minimum impact on the performance of the bouquet and provide robust upper bounds on MSO induced thereof. This is to minimize the effect of the increase in sub-optimality for bouquet

4. The approach should be generic, independent of specificities of SQL language, or any other, and extend-able to any query predicate or combination of predicates.

3 Algorithm

We will refer Ikebana Bouquet and also the Algorithm (3) that generates the plan bouquet for reduced ESS as Ikebana.However the interpretation will be clear from the context The Ikebana Algorithm finds the redundant dimension according to the definition given in section 2.2.The algorithm takes as input the cost of all plans at all points in the grid.Along with the dimension index which can be reduced , algorithm output is the set of plan in reduced space and their associated budget.

Algorithm 1 Ikebana Algorithm

```
Algo-Ikebana(planCost,dimension,resolution)
for each dimension $d$
 for each hyperplane $h$ in $d$
  for all the points $p$ modulo points on $h$
    find the optimal plan , from the set of plans on $h$, at point $p$
  find maxDiffPair = (bestCost in reduced ESS,optCost)
  such that (bestCost in reduced ESS-optCost) is max $\forall$ points
  find (min,max) cost for all plans on the hyperplane $h$
  find $\rho$ , $k'$ and $k$ using maxDiff , using r = 2, in reduced ESS   find $\delta = k' - k$
  calculate $MSO_h$ for the hyperplane $h$ given by (7)
  choose the hyperplane with the least $MSO_h$ return ($h^*$,$MSO^*_h$,(set of plans on $h^*$,min,max))
```

The algorithm works as follows:
On a fixed dimension d , we find all optimal plans ,fixedHyperPlanePlans, on a hyperplane of dimension D - 1.Consider fixedHyperPlanePlans as the reduced set of POSP plans.For all the points on grid we find optimal cost of execution using only the reduced set of POSP plans.For fixedHyperPlanePlans , $C^{'}_{min}$ and $C^{'}_{max}$, are calculated to find the MSO given by (7).For current fixedHyperPlane ,find ρ and maximum $\delta = k' - k$,where k' is the contour of execution of reduced bouquet and k is the contour of actual query location.Hyperplane which gives the lowest MSO is the best and will be chosen ,along with the optimal set of plans on it and their min and max budget ,provided the current dimension,d, is reducible.Dimension d is reduced if it gives the lowest MSO among

all the minimum MSO found when all dimensions are compared. Each dimension is checked for reducibility and the one which gives lowest sub-optimality is eschewed.There is no preset criteria to decide which dimension is to be removed.We find the reduced MSO,given by (7) , after removing each dimension separately.The dimension which leads to lowest reduced MSO is removed.
The algorithm also outputs overlap factor(for each POSP plans in the reduced ESS) which is defined as follows: $OverlapFactor_i$ is defined as the ratio of number of points at which a plan i executes at run time due to increased budget to the number of points at which plan i is best in reduced ESS,i.e. spill of plan i at runtime.

Complexity :Time complexity of the proposed algorithm is exponential in dimension,given by $O(rDr^D)$,where r is the resolution of the grid and D is the original number of dimensions.The reason for this high complexity is that we have to explore all the possibilities by checking each dimension individually whether it is reducible.Note that this is only one time process which can be done at compile time and at run time, Bouquet ,on reduced set of dimensions, can be used to process query.

4 Ikebana Analysis

In this section we define the sub-optimality of Ikebana Bouquet and derive an expression for the same.Ikebana Bouquet is essentially the Plan Bouquet approach but on reduced ESS.We compare Ikebana performance w.r.t. that of an oracle and as a special case show comparison w.r.t. Plan Bouquet on original ESS.Once some subset of dimensions have been reduced, Ikebana PIC curve(in reduced ESS) will be above the Plan Bouquet's PIC(without reduction in ESS) curve. Traditional assumptions of PCM holds here as well, i.e. PIC is an increasing function and also continuous throughout ESS. Minimum and Maximum costs for Ikebana are denoted by C'_{min} and C'_{max}, respectively and C_{min} and C_{max}, are minimum and maximum costs , respectively on original ESS.Now consider the case wherein the isocost steps are organized in a geometric progression with initial value a ($a > 0$) and common ratio r ($r > 1$).PIC is sliced with $m = log_r \left\lceil \frac{C'_{max}}{C_{min}} \right\rceil$ cuts, $IC_1, ..., IC_m$, satisfying the boundary conditions $a/r < C_{min} \leq IC_1$ and $a > C_{min}$. Sub-optimality of Ikebana, MSO_I ,for some query q which lies on or below the k^{th} contour, w.r.t. the 'oracle' algorithm is defined as the ratio of cost of Ikebana,$C_I(q_{k'})$, to the optimal cost ,$C^*(q_k)$. Thus we have

$$C_I(q_{k'}) = cost(IC_1) + ... + cost(IC_{k'}) = \frac{a(r^{k'} - 1)}{r - 1} \tag{1}$$

$$C^*(q_k) = ar^{k-2} \tag{2}$$

We have skipped the full derivation for the costs given above as user can refer section 3.1 in [Anshuman and Jayant2014]. Note that (1) has k' and (2) has k as the index of the contour.That's because Ikebana executes on reduced POSP which induces sub-optimality but 'oracle' algorithm magically knows the actual query location on original ESS and may not $k' - 2$.

Theorem 1. *Given a query Q with multi-dimensional ESS S where $S = \{d_1, ..., d_s\}$ the associated PIC discretized with a geometric progression having common ratio r , maximum contour plan density ρ for Ikebana ,C_{max} and C_{min} be the maximum and minimum cost, respectively on S_r ,where S_r is the reduced space, $S_r = \{S \setminus d_k\}$ for some k , after one dimension d_k has been reduced.Then the MSO for Ikebana w.r.t. oracle algorithm is given by*

$$MSO_I \leq \frac{\rho r^{\delta+2}}{r-1} \tag{3}$$

Proof. Using (1) and (2) and from the definition of MSO_I , we get

$$MSO_I = \frac{\frac{a(r^{k'}-1)}{r-1}}{ar^{k-2}} \tag{4}$$

Let k' be given as

$$k' = k + \delta \tag{5}$$

Using (4) and (5) , we get

$$MSO_I = \frac{a(r^{k+\delta}-1)}{(r-1)(ar^{k-2})} = \frac{r^{\delta+2}}{r-1} - \frac{r^{2-k}}{r-1} \leq \frac{r^{\delta+2}}{r-1} \tag{6}$$

Note that δ depends on the actual query location at run time.
For *dimensions* $>= 2$ in reduced ESS , there will be a ρ term in (6) such that

$$MSO_I \leq \frac{\rho r^{\delta+2}}{r-1} \tag{7}$$

where ρ is the number of plans in the densest contour of Ikebana. □

Note that when original ESS dimension is 1 , $\rho = 1$ and there is no scope of dimension reduction

Lemma 1. *Taking r = 2 in Theorem 1 , we get*

$$MSO_I \leq 4\rho 2^{\delta}$$

Note the presence of 4ρ in the above lemma which is the expression for MSO for Plan Bouquet as given by Theorem 3 in [Anshuman and Jayant2014].

5 Experiments

We note here that the problem of dimension reduction is embarrassingly parallel in both : exploring a given dimension and exploring each HyperPlane within the dimension(to be reduced) as candidates for reduction and exploit this observation in the Ikebana implementation. The Ikebana algorithm(given by Algorithm 1) was implemented in Python2.7 and the source is available for inspection[1].Ikebana forks one thread for each dimension to reduce and explores this space in parallel reducing the time of exponential execution by several magnitudes. The Ikebana algorithm was run on the TPC-H benchmark query Q5.

[1]https://github.com/adarshpatil/ESS-Reduction

This query has 3 ESS dimensions and was sampled for a 20x20x20 grid. The database optimizer returned 50 POSPs for this ESS. The code was run on a 8 core AMD CPU, 16GB RAM system and the exploration for the above mentioned query completed execution in 5.3 seconds. The adjoining figures show the range of MSO values the bouquet execution will encounter depending on the point of the query delta values for the reduced POSP in chosen row. As proposed earlier in section 3 Ikebana executes and chooses the HyperPlane that gives the least value of MSO.

Based on this, the Ikebana implementation outputs the bouquet plans, min and max Cost and overlap factor using the Theorem 1

Figure 1: Min/Max MSO bounds for Reducing dimension 1 on TPC-H Q5

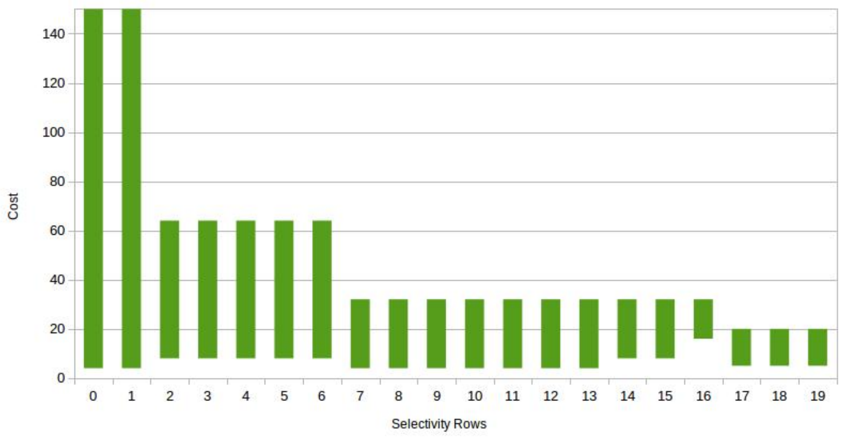


Figure 1 above shows the min and max bounds on execution of Plan Bouquet algorithm for the plans output by Ikebana. The corresponding plan bouquet and the min, max budgets are seen in Table 1, Column 1 below. We see the graded difference in scales of bounds between the first/second HyperPlanes (which extend beyond the scale of the graph) and the following HyperPlanes. This shows that the plans that are found on lower selectivity rows do not perform well as selectivity increases and cannot handle the removal of dimensions. On the other hand plan, HyperPlanes 17 shows good upper and lower bounds on the MSO and are able to handle the removal of dimension well. Thus this dimension is chosen for reduction. Also, we observe a tie in MSO between HyperPlanes 17, 18 and 19. This tie is broken on the basis of average overlap factor for the HyperPlane as defined in Section 3 Similarly Figure 2 and Figure 3 shows bounds for removal of dimension 0 and dimension 2 respectively and the bouquet of plans along with min and max budget are show in Column 2 and Column 3 of the Table 1.

Table 1: Ikebana Bouquet for reduced dimension with budgets for TPC-H Q5

```
Reducing dimension 0
Use selectivity row 5 with MSO 24.0

Here's the Plan Bouquet
Plan Number : Min Cost : Max Cost : Overlap Factor
0 : 32047.57154 : 72584.8408 : 1.0
33 : 143221.48723 : 271056.00391 : 28.0
2 : 39315.74284 : 149520.57702 : 9.0
35 : 233741.21581 : 594256.69581 : 2.0
32 : 163600.859204 : 383326.749728 : 58.0
34 : 39316.88784 : 272311.44922 : 2.0
19 : 87052.302142 : 248789.973949 : 2.0
24 : 94099.625582 : 251011.807459 : 5.0
27 : 143187.97099 : 143196.36005 : 60.0
28 : 39316.01034 : 149520.73702 : 38.0
30 : 39316.25034 : 195476.54914 : 14.0
```

```
Reducing dimension 1
Use selectivity row 17 with MSO 20.0

Here's the Plan Bouquet
Plan Number : Min Cost : Max Cost : Overlap Factor
32 : 163600.859204 : 251555.591209 : 55.0
2 : 32514.30034 : 150061.09922 : 2.0
3 : 81292.75778 : 87647.38038 : 3.0
4 : 81334.47808 : 87647.58788 : 13.0
6 : 81392.754 : 87648.12038 : 6.0
8 : 81501.09053 : 87656.48038 : 5.0
41 : 234086.10581 : 464448.93956 : 4.0
42 : 286990.602609 : 330700.914017 : 27.0
43 : 315186.65456 : 498885.98956 : 103.0
34 : 34713.21409 : 272311.44922 : 2.0
35 : 233741.21581 : 295951.06331 : 5.0
24 : 94054.096832 : 251011.807459 : 2.0
28 : 32786.86034 : 173532.76672 : 15.0
30 : 33062.96784 : 195476.54914 : 7.0
```

```
Reducing dimension 2
Use selectivity row 3 with MSO 32.0

Here's the Plan Bouquet
Plan Number : Min Cost : Max Cost : Overlap Factor
0 : 32047.57154 : 72582.7633 : 2.0
2 : 39315.74284 : 150061.09922 : 9.0
3 : 81292.75778 : 87688.14913 : 1.0
44 : 32344.90764 : 72583.0533 : 16.0
34 : 39316.88784 : 272311.44922 : 2.0
46 : 32654.04994 : 72583.5583 : 11.0
15 : 87051.727142 : 248841.642699 : 4.0
35 : 233741.21581 : 594256.69581 : 2.0
24 : 94099.625582 : 384929.793478 : 6.0
47 : 34450.56499 : 72584.56205 : 21.0
28 : 39316.01034 : 173532.76672 : 40.0
30 : 39316.25034 : 195476.54914 : 14.0
```

Figure 2: Min/Max MSO bounds for Reducing dimension 0 on TPC-H Q5

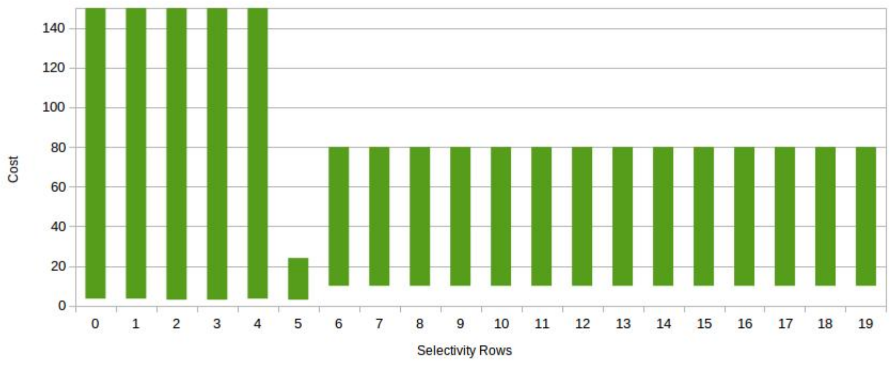
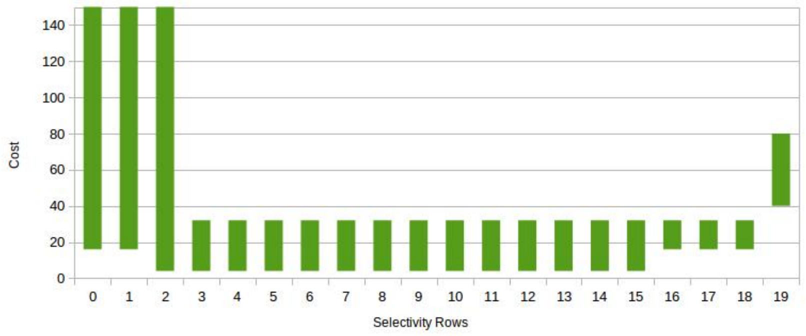


Figure 3: Min/Max MSO bounds for Reducing dimension 2 on TPC-H Q5



6 Conclusion and Future Work

Ikebana has successfully quantified the impact of reducing dimensions in ESS through experiments on TPC-H benchmarks and established concrete bounds on suboptimality.The approach given in the report can be used to improve bouquet performance and reduce time consumed at run time due to high plan density and large number of error prone predicates.Ikebana reduces compile time cost for Plan Bouquet as the number of dimensions to explore are reduced. The impact of overlap factor defined in Section 3 can be studied and analyzed further and the number of points for which multiple plans exist can be reduced.Ikebana Approach can be extended to reduce multiple dimensions by following iterative scheme such that redundant dimensions are removed one by one till MSO reaches some upper limit.Thus future work can be in the direction of exploring whether multiple dimensions can be reduced iteratively one at a time or a combination of few dimensions taken at once yields better results

References

[Anshuman and Jayant2014] Anshuman Dutt and Jayant R. Haritsa. 2014. *Plan bouquets: Query Processing without Selectivity Estimation. ACM SIGMOD Conf. 2014.*