Cost-based Query Optimization

Declarative SQL Query (Q) → Query Optimizer (dynamic programming) → Minimum Cost Execution Plan P(Q)

Search Space → Cost Model → DB catalogs

Cost of Nested Loops Block-Join of R and S = |R| + |R| * |S|

Num rows/blocks in relation R
Num unique values in attribute A

April 2011 Plan Diagrams Tutorial (ICDE 2011)
select StudentName, CourseName
from STUDENT, COURSE, REGISTER
where STUDENT.RollNo = REGISTER.RollNo
    and REGISTER.CourseNo = COURSE.CourseNo
    and REGISTER.date < 2000
    and COURSE.credits < 2
RETURN
Cost: 286868

Total Execution Cost (estimated)
Relational Selectivities

- Cost-based Query Optimizer’s choice of execution plan = \( f(\text{query, database, system, \ldots}) \)

- For a given database and system setup, execution plan chosen for a query = \( f(\text{selectivities of query’s base relations}) \)
  - selectivity is the estimated percentage of rows of a relation used in producing the query result
Determines the values of goods shipped between nations in a time period

```sql
select supp_nation, cust_nation, l_year, sum(volume) as revenue
from (select n1.n_name as supp_nation, n2.n_name as cust_nation, extract(year from l_shipdate) as l_year, l_extendedprice * (1 - l_discount) as volume
from supplier, lineitem, orders, customer, nation n1, nation n2
where s_suppkey = l_suppkey and o_orderkey = l_orderkey and c_custkey = o_custkey and s_nationkey = n1.n_nationkey and c_nationkey = n2.n_nationkey and ((n1.n_name = 'FRANCE' and n2.n_name = 'GERMANY') or (n1.n_name = 'GERMANY' and n2.n_name = 'FRANCE')) and l_shipdate between date '1995-01-01' and date '1996-12-31'
and o_totalprice <= C1 and c_acctbal <= C2) as shipping
group by supp_nation, cust_nation, l_year
order by supp_nation, cust_nation, l_year
```
Plan and Cost Diagrams

- A plan diagram is a pictorial enumeration of the plan choices of the query optimizer over the relational selectivity space.

- A cost diagram is a visualization of the (estimated) plan execution costs over the same relational selectivity space.
Sample Plan Diagram
[QT7, OptB, Res=100]
Sample Cost Diagram
[QT7, OptB, Res=100]

MinCost: 6.08E3
MaxCost: 3.24E4
TUTORIAL OUTLINE

Part I: Plan Diagram Characteristics [VLDB 2005]

Part II: Plan Diagram Production [VLDB 2005/2008]

Part III: Plan Diagram Reduction [VLDB 2007]

Part IV: Robust Plan Diagrams [VLDB 2008]

Part V: Intra-optimizer Integration [VLDB 2010]

Part VI: Future Research Directions
Picasso Visualizer

Picasso is a (free) Java tool that, given an $n$-dimensional SQL query template and a choice of database engine, automatically generates plan and cost diagrams

- Operational on
  - DB2
  - Oracle
  - SQLServer
  - Sybase
  - PostgreSQL
  - MySQL

- Additional Diagrams:
  - Cardinality Diagram
  - Plan-tree Diagram
  - Plan-difference diagram
  - Abstract-plan diagram
  - ....

DEMO
Testbed Environment

- **Benchmark Databases**
  - TPC-H (1 GB)
  - TPC-DS (100 GB)

- **Query Templates**
  - 2-D, 3-D, 4-D query templates based on TPC-H [Q1 ~ Q22] and TPC-DS [Q1 ~ Q99] query suites

- **Relational Engines**
  - Default installations (with all optimization features on)
  - Statistics on all the parametrized attributes

- **Computational Platform**
  - Vanilla PC/Workstation

<table>
<thead>
<tr>
<th>TPC-H Relation</th>
<th>Relation Cardinality</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGION</td>
<td>5</td>
</tr>
<tr>
<td>NATION</td>
<td>25</td>
</tr>
<tr>
<td>SUPPLIER</td>
<td>10000</td>
</tr>
<tr>
<td>CUSTOMER</td>
<td>150000</td>
</tr>
<tr>
<td>PART</td>
<td>200000</td>
</tr>
<tr>
<td>PARTSUPP</td>
<td>800000</td>
</tr>
<tr>
<td>ORDERS</td>
<td>1500000</td>
</tr>
<tr>
<td>LINEITEM</td>
<td>6001215</td>
</tr>
</tbody>
</table>
The Picasso Connection

Plan diagrams are often similar to cubist paintings!

[ Pablo Picasso – founder of cubist genre ]

Woman with a guitar
Georges Braque, 1913
Smooth Plan Diagram
[QT7, OptB, Res=100]
Highly irregular plan boundaries

Intricate Complex Patterns

The Picasso Connection

Extremely fine-grained coverage (P76 ~ 0.01%)

Gini Index: 0.83

Increases to 90 plans with 300x300 grid!
Cost Diagram

[QT8, Opt A*, Res=100]

All costs are within 20% of the maximum

MinCost: 8.26E5
MaxCost: 1.05E6
# Plan Space Coverage

![Diagram](https://example.com/diagram.png)

**80-20 Rule**

Gini skew index $> 0.5$

<table>
<thead>
<tr>
<th>TPCH Query Template</th>
<th>Opt A</th>
<th>Opt B</th>
<th>Opt C</th>
</tr>
</thead>
<tbody>
<tr>
<td>QT2</td>
<td>Plan</td>
<td>80%</td>
<td>Gini</td>
</tr>
<tr>
<td></td>
<td>Cardinality Coverage</td>
<td>Index</td>
<td></td>
</tr>
<tr>
<td>QT2</td>
<td>31</td>
<td>25</td>
<td>38</td>
</tr>
<tr>
<td>QT5</td>
<td>63</td>
<td>44</td>
<td>41</td>
</tr>
<tr>
<td>QT7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QT8</td>
<td>5</td>
<td>13</td>
<td>5</td>
</tr>
<tr>
<td>QT9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QT10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QT18</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QT21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Picasso Art Gallery

- Duplicates and Islands
- Plan Switch Points
- Venetian Blinds
- Footprint Pattern
- Speckle Pattern
Violates basic tenets of *Parametric Query Optimization* (PQO) literature:

- **Plan Convexity**: Plan optimal at $X$ and $Y$, is also optimal at all locations on the line joining $X$ and $Y$;
- **Plan Uniqueness**: An optimal plan appears at only a single contiguous region in the space;
- **Plan Homogeneity**: An optimal plan is optimal within the entire region enclosed by its boundaries.
Plan Switch Point:
line parallel to axis with a plan shift for all plans bordering the line.

Hash-Join sequence
PARTSUPP \Join \sup \JOIN \Join SUPPLIER \Join PART
is altered to
PARTSUPP \Join PART \Join SUPPLIER

April 2011
Plan Diagrams Tutorial (ICDE 2011)
Six plans simultaneously change with rapid alternations to produce a “Venetian blinds” effect.

Left-deep hash join across NATION, SUPPLIER and LINEITEM relations gets replaced by a right-deep hash join.
P7 is a thin and broken curved pattern in the middle of P2's region.

P2 has sort-merge-join at the top of the plan tree, while P7 uses hash-join.
An additional sort operation is present on the PART relation in P2, whose cost is very low.
Non-Monotonic Cost Behavior

- Plan-Switch Non-Monotonic Costs
- Intra-Plan Non-Monotonic Costs
Plan-Switch Non-Monotonic Costs

[QT2, OptA]

26%
Selectivity

50%
Selectivity

26%: Cost decreases by a factor of 50
50%: Cost increases by a factor of 70
Intra-Plan Non-Monotonic Costs

[NQT21,OptA]

Nested loops join whose cost decreases with increasing input cardinalities

Plan Diagram

Cost Diagram
Remarks

- Modern optimizers tend to make extremely fine-grained and skewed choices
  - an over-kill, not merited by the coarseness of the underlying cost space
  - collateral damage of becoming too complex over time, making it difficult to anticipate module interactions

- Is it feasible to reduce the plan diagram complexity without materially affecting the plan quality? [PART III of Tutorial]
TUTORIAL OUTLINE

Part I: Plan Diagram Characteristics [VLDB 2005]

Part II: Plan Diagram Production [VLDB 2005/2008]

Part III: Plan Diagram Reduction [VLDB 2007]

Part IV: Robust Plan Diagrams [VLDB 2008]

Part V: Intra-optimizer Integration [VLDB 2010]

Part VI: Future Research Directions
Diagram Generation Process

Diagram showing the relationship between ORDERS.o_totalprice and CUSTOMER.c_acctbal.
Diagram Generation Overheads

- Generating a 2D plan diagram at resolution 1000, or 3D at resolution 100, requires $10^6$ optimizations
- Cost of each optimization: $\sim 0.5$ sec
- Running time: $\sim 1$ WEEK!
Research Challenge

Can we obtain an accurate approximation in reasonable time?
Approximation Metrics

- **Notation**
  - $P$: true plan diagram
  - $A$: approximate plan diagram
  - $|P|$ and $|A|$: number of plans present in $P$ and $A$, respectively
  - $p_P(q)$ and $p_A(q)$: plans assigned to query point $q$ in $P$ and $A$, respectively
  - $m$: number of query points in the diagrams

- **Plan Identity Error ($\mathcal{E}_I$):** % of plans that remained unidentified in $A$ relative to $P$
  \[
  \mathcal{E}_I = \frac{|P| - |A|}{|P|} \times 100
  \]

- **Plan Location Error ($\mathcal{E}_L$):** % of points assigned wrong plan in $A$ relative to $P$
  \[
  \mathcal{E}_L = \frac{|p_A(q) \neq p_P(q)|}{m} \times 100
  \]
Highly irregular optimality boundaries \Rightarrow \text{Difficult Plan Location Error}

Large number of very small plans \Rightarrow \text{Difficult Plan Identity Error}
SOLUTION TECHNIQUES

- Purely Statistical:
  - Random Sampling with Nearest Neighbor Inferencing

- DB-conscious:
  - Grid Sampling with Parametric Query Optimization (GS_PQO)
Basic Grid Sampling

- Partition Selectivity Space into coarse grid, optimize corners.
- Process middle points of each edge
  - If end points have the same plan, assign this plan to the middle point also
  - Else explicitly optimize the point
- Process center of each rectangle
  - Check end points of the crosshairs
  - If either pair of ends have a common plan, assign this plan to the center
  - Else explicitly optimize the point
- Iteratively partition until 1x1 box
  (i.e. all points in the selectivity space have been processed).
Micro-PQO heuristic

- PQO principle: If two points in a query parameter space have the same optimal plan, then this plan is optimal at all points on the straight line joining them.

- Plan Diagrams severely violate PQO [Part I]
- But, PQO usually holds in micro-regions
Issues with Basic Grid Sampling

- Rectangles that are similar w.r.t. corners may internally have different plan richness

  ![Diagram showing two rectangles with different inner structures.]

- Treated as same by Grid Sampling approach

- Samples should be assigned \( \propto \) Plan Density
Observation

Matching nodes are colored white
Conjecture

Morphing of one plan tree to other occurs in incremental steps.

- More structural difference in Plan Trees
- Increased Plan Density
- More points need to be optimized

- Therefore plan tree difference can be used as an indicator of “Plan Density”
Quantifying Plan Difference

- Use classical Jaccard Distance metric
- Let plan trees $T_i$ and $T_j$ have $|T_i|$ and $|T_j|$ nodes, respectively, and $|T_i \cap T_j|$ denote the number of matching nodes between them. Then, Plan Density factor is estimated as

$$\rho = 1 - \frac{T_i \cap T_j}{T_i \cup T_j}$$

- Hyper-rectangle with $n$ corner points and plan trees $T_1, T_2 \ldots T_n$. Then, overall Plan Density factor is estimated as

$$\rho(T_1, T_2 \ldots T_n) = \frac{\sum_{i=1}^{n} \sum_{j=i+1}^{n} \rho(T_i, T_j)}{\binom{n}{2}}$$
Plan Density Example

$\rho$ is a metric normalized to $[0,1]$

- $\rho$ close to 0 indicates similar plan trees
- $\rho$ close to 1 indicates extremely dissimilar plan trees

Indication of high plan density

Indication of low plan density
Complete GS_PQO

- Optimize the initial grid rectangles
- Calculate $\rho$ for each box and Insert in heap
- MAX HEAP
  - Extract box with highest $\rho$
  - $\rho > 0.1$
    - Yes: Process the box and divide further
    - No: Infer remaining points in the box
  - Insert

---

April 2011

Plan Diagrams Tutorial (ICDE 2011)
Complex Diagram Approximation Examples

GS_PQO

Sample size 7%  \( \varepsilon_I, \varepsilon_L < 10\% \)

Sample size 15%  \( \varepsilon_I, \varepsilon_L < 10\% \)
TUTORIAL OUTLINE

Part I: Plan Diagram Characteristics [VLDB 2005]
Part II: Plan Diagram Production [VLDB 2005/2008]
Part III: Plan Diagram Reduction [VLDB 2007]
Part IV: Robust Plan Diagrams [VLDB 2008]
Part V: Intra-optimizer Integration [VLDB 2010]
Part VI: Future Research Directions
Can the plan diagram be recolored with a smaller set of colors (i.e. some plans are “swallowed” by others), such that

**Guarantee:**

*No query point in the original diagram has its estimated cost increased, post-swallowing, by more than $\lambda$ percent* (user-defined)

**Analogy:**

*Cuba agrees to be annexed by USA if it is assured that the cost of living of each Cuban citizen is not increased by more than $\lambda$ percent*
Reduced Plan Diagram $[\lambda=10\%]$  
[QT8, OptA*, Res=100]

Comparatively smoother contours

Reduced to 5 plans from 76!
Definition

- Plan diagram $P$
  - $m$ query points $q_1 \ldots q_m$
  - $n$ optimal plans $P_1 \ldots P_n$
- Each query point $q_i$
  - Selectivity location $(x\%, y\%)$
  - Cost of plan $P_j$ at $q_i$ is $c(P_j, q_i)$
  - Optimal plan $P_k \Rightarrow$ Color $L_k$
- Cost-increase threshold $\lambda\%$ (user defined)
- Reduced plan-diagram $R$
  - $L^R \subseteq L^P$

Problem: Find an $R$ such that the number of plans (colors) in $R$ is minimum subject to

\[ \forall P_k \in P, \text{ either} \]
\[ (a) P_k \in R \text{ or} \]
\[ (b) \forall q \in P_k, \text{ the assigned replacement plan } P_j \in R \text{ is} \]
\[ s.t. \quad \frac{c(P_j, q)}{c(P_k, q)} \leq 1 + \frac{\lambda}{100} \]

e.g. if $\lambda = 10\%$, \[ \frac{c(P_j, q)}{c(P_k, q)} \leq 1.1 \]
Basic Requirement

- Need to be able to cost a plan $P_k$ at points outside its own optimality region, called “Foreign Plan Costing” (FPC)

- Option 1:
  - some optimizers natively support FPC feature
  - incurs non-trivial computational overheads

- Option 2:
  - use a conservative cost-upper-bounding approach
  - orders of magnitude faster
Option 2 Assumption: Plan Cost Monotonicity (PCM)

PCM: Cost distribution of each plan featured in plan diagram $P$ is monotonically non-decreasing over entire selectivity space $S$.

True for most query templates since

selectivity $\uparrow \Rightarrow$ input data $\uparrow \Rightarrow$ query processing $\uparrow \Rightarrow$ (est) cost $\uparrow$
Cost-upper-bounding Approach

PCM ⇒
Cost of a “foreign” query point in first quadrant of $q_s$ is an upper bound on the cost of executing the foreign plan at $q_s$

⇒
Cost of executing $q_s$ with foreign plans $P_1$ or $P_4$ lies in the intervals $[88, 90]$ and $[88, 91]$, respectively.

Cost of query point $q_s$ with optimal plan $P_2$ is 88
Example Plan Swallowing

\[ \lambda = 10\% \]

\[ \text{ORDERS} \]

\[ \text{CUSTOMER} \]

\[ (100, 103) \]
Results

- **Optimal plan diagram reduction (w.r.t. minimizing the number of plans/colors) is NP-hard**
  - through problem-reduction from classical Set Cover

- **Designed CostGreedy, a greedy heuristic-based algorithm with following properties:**
  [m is number of query points, n is number of plans in diagram]
  - **Time complexity is** \( O(mn) \)
    - linear in number of plans for a given diagram resolution
  - **Approximation Factor is** \( O(\ln m) \)
    - bound is both tight and optimal
    - in practice, closely approximates optimal
Cost Greedy Algorithm

- Assign a bin to each individual plan in $P$
- Start at the top right corner and proceed in reverse row-major order
  - first-quadrant info available when processing a query point
- Put a copy of each query point into all plan-bins (subsets) that it can belong to w.r.t. $\lambda$ constraint: SetCover problem
- Iterative Greedy Criterion:
  - include in solution the plan (subset) that covers the maximum number of uncovered points
  - remove its covered points from all subsets and repeat until no uncovered points remain
Toy Example

Plans in R

- Green
- Yellow

P

R

- Y3
- Y2
- C1

- R4
- R5
- B6

- B9
- R8
- R7

Pick this plan
Covers max (3) points
Anorexic Reduction

Extensive empirical evaluation with a spectrum of multi-dimensional TPC-H and TPC-DS based SQL query templates indicates that

“With a cost-increase-threshold of just 20%, virtually all complex plan diagrams [irrespective of query templates, data distribution, query distribution, system configurations, etc.] reduce to “anorexic levels” (~10 or less plans)!”
## Sample Reduction Results

\[ \text{OptC, Res = 30E, } \lambda = 20\% \]

<table>
<thead>
<tr>
<th>TPC-H Query Template</th>
<th>Original # of Plans</th>
<th>Reduced Plans CostGreedy</th>
<th>Reduced Plans CG-FPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>QT2</td>
<td>60</td>
<td>14</td>
<td>3</td>
</tr>
<tr>
<td>QT5</td>
<td>51</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>QT8</td>
<td>121</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>QT9</td>
<td>137</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>QT10</td>
<td>44</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
Applications of Anorexic Plan Diagram Reduction

- Quantifies redundancy in plan search space
- Provides better candidates for plan-cacheing
- Enhances viability of Parametric Query Optimization (PQO) techniques
- Improves efficiency/quality of Least-Expected-Cost (LEC) plans
- Minimizes overheads of multi-plan (e.g. Adaptive Query Processing) approaches
- Identifies selectivity-error resistant plan choices
  - retained plans are robust choices over larger regions of the selectivity space
TUTORIAL OUTLINE

Part I: Plan Diagram Characteristics [VLDB 2005]
Part II: Plan Diagram Production [VLDB 2005/2008]
Part III: Plan Diagram Reduction [VLDB 2007]
Part IV: Robust Plan Diagrams [VLDB 2008]
Part V: Intra-optimizer Integration [VLDB 2010]
Part VI: Future Research Directions
Selectivity Estimation Errors

\[ q_e(x_e, y_e) : \text{estimated location by optimizer} \]
\[ q_a(x_a, y_a) : \text{actual location during execution} \]

The difference could be substantial due to
- Outdated Statistics (expensive to maintain)
- Coarse Summaries (histograms)
- Attribute Value Independence (AVI) assumptions
Impact of Error Example

Estimated Query Location ($q_e$): (1, 40)

Actual Query Location ($q_a$): (80,80)

Cost($P_{oe}$) = $9 \times 10^4$ (optimal)

Cost($P_{oa}$) = $50 \times 10^4$ (optimal)

Cost($P_{oe}$) = $110 \times 10^4$ (highly sub-optimal)

Selectivity Space $S$
Error Locations wrt Plan Replacement Regions

Inherently robust

Endo-optimal_{re}

Swallow_{re}

Exo-optimal_{re}
Positive Impact of Reduction

In most cases, replacement plan provides robustness to selectivity errors even in exo-optimal region

\[ q_e = (0.36, 0.05) \]

Original Plan \( (P_{oe}) \)
Replacement Plan \( (P_{re}) \)
Local Optimal Plan \( (P_{oa}) \)
Negative Impact of Reduction

But, occasionally, the replacement is much worse than the original plan!

QT5
$q_e = (0.03, 0.14)$

Replacement Plan ($P_{re}$)
Original Plan ($P_{oe}$)
Local Optimal Plan ($P_{oa}$)
Research Challenge

How do we ensure that plan replacements can only help, but never materially hurt the expected performance?
Globally Safe Replacement

- Earlier local constraint:
  \( P_{re} \) can replace \( P_{oe} \) if
  \[
  \forall \text{ points } q \text{ in } P_{oe}'s \text{ endo-optimality region},
  c(P_{re}, q) \leq (1 + \lambda) c(P_{oe}, q)
  \]

- New global constraint:
  \( P_{re} \) can replace \( P_{oe} \) only if it guarantees a globally safe space
  \[
  \forall \text{ points } q \text{ in selectivity space } S,
  c(P_{re}, q) \leq (1 + \lambda) c(P_{oe}, q)
  \]
Globally Safe Replacement

\[ \text{Safe} \left( P_{re}, P_{oe} \right) \]
USA can annex Cuba only if American passport can guarantee cost of living of Cuban citizen is within $\lambda$ of that obtained with the Cuban passport, irrespective of the country to which the Cuban citizen emigrates.
Solution Strategy

- **Foreign Plan Costing (FPC)** feature is mandatory
- Characterize behavior of all plans throughout the selectivity space $S$ using FPC
- Not a viable solution in practice
  - Requires $O(mn)$ FPC to be performed $[10^6 \leftrightarrow 10^9]$ 
    - $m$: number of query points; $n$: number of optimal plans
  - Although costing cheaper than optimization (1:10), the sheer number makes it prohibitively expensive

*Can we reduce the number of FPC invocations to a manageable extent?*
Plan Cost Model (2D)

Given selectivity variations \( x \) and \( y \), for any plan \( P \) in the plan diagrams of current optimizers, we can fit:

\[
\text{PlanCost}_P(x,y) = a_1 x + a_2 y + a_3 xy + a_4 x \log x + a_5 y \log y + a_6 xy \log xy + a_7
\]

The specific values of \( a_1 \) through \( a_7 \) are a function of \( P \).

Extension to \( n \)-dimensions is straightforward.
Cost Model Fit Example

Original Cost Function

\[ \text{Cost}(x, y) = 17.9x + 45.9y + 1046xy - 39.5x \log x + 4.5y \log y + 27.6xy \log xy + 97.3 \]

Fitted Cost Function
Main Result

Given the 7-coefficient plan cost model, need to perform FPC at only the perimeter of the selectivity space to determine global safety

Border Safety $\Rightarrow$ Interior Safety!
Safe and Violating Points

- $f_{oe}(x,y)$: cost function of $P_{oe}$
- $f_{re}(x,y)$: cost function of $P_{re}$

Safety Function

$$f(x,y) = f_{re}(x,y) - (1 + \lambda) f_{oe}(x,y)$$

Wrt this replacement,

- $q$ is a safe point if $f(x_q, y_q) \leq 0$
- $q$ is a violating point if $f(x_q, y_q) > 0$

Globally Safe Space – no violating points in entire selectivity space
Safety Function Behavior

Assume both (d) and (e) are unsafe

Checked through first and second derivatives

SC1

SC2

SC3
Safety Check Theorem

For a plan pair \((P_{oe}, P_{re})\) and a selectivity space \(S\) with corners \([(x_1,y_1), (x_1,y_2), (x_2,y_2), (x_2,y_1)]\), the replacement is safe in \(S\) if any one of the conditions \(SC1\) through \(SC6\) is satisfied.

<table>
<thead>
<tr>
<th></th>
<th>Left Boundary</th>
<th>Right Boundary</th>
<th>Top Boundary</th>
<th>Bottom Boundary</th>
</tr>
</thead>
<tbody>
<tr>
<td>(SC1)</td>
<td>Safe</td>
<td>Safe</td>
<td>(f''_{y_2}(x) \geq 0)</td>
<td>(f''_{y_1}(x) \geq 0)</td>
</tr>
<tr>
<td>(SC2)</td>
<td>(f_{y_1}(x_1) \leq 0) &amp; Safe</td>
<td>Safe</td>
<td>(f''_{y_2}(x) &lt; 0)</td>
<td>(f''_{y_1}(x) &lt; 0)</td>
</tr>
<tr>
<td>(SC3)</td>
<td>Safe</td>
<td>(f_{y_2}(x_2) \geq 0) &amp; Safe</td>
<td>(f''_{y_2}(x) &lt; 0)</td>
<td>(f''_{y_1}(x) &lt; 0)</td>
</tr>
<tr>
<td>(SC4)</td>
<td>(f_{x_1}(y) \geq 0)</td>
<td>(f''_{x_2}(y) \geq 0)</td>
<td>Safe</td>
<td>Safe</td>
</tr>
<tr>
<td>(SC5)</td>
<td>(f_{x_1}(y) &lt; 0)</td>
<td>(f''_{x_2}(y) &lt; 0)</td>
<td>(f_{x_2}(y_2) \geq 0) &amp; Safe</td>
<td>Safe</td>
</tr>
<tr>
<td>(SC6)</td>
<td>(f''_{x_1}(y) &lt; 0)</td>
<td>(f''_{x_2}(y) &lt; 0)</td>
<td>Safe</td>
<td>(f''_{x}(y_1) \leq 0) &amp; Safe</td>
</tr>
</tbody>
</table>
Safety Check Algorithm

Perimeter Test
SC2, SC3, SC5 & SC6

Wedge Test
SC1 & SC4
SEER [Selectivity Estimation Error Resistance]

Plan Replacement Algorithm

- Create a Set Cover instance $I = (U, S)$
  - $U = \{1, 2, \ldots, n\}$, $S = \{S_1, S_2, \ldots, S_n\}$
  - $S_i = \{i\}$, $i = \{1, \ldots, n\}$
- For each pair of plans $(P_i, P_j)$
  - If $P_i$ can “safely swallow” $P_j$, then $S_i = S_i \cup \{j\}$ (using the GlobalSafetyCheck routine)
- Solve (using Greedy SetCover) the Set Cover instance to obtain the reduced plan diagram
Error Resistance Example

Original (Aggregate) Cost Diagram

Provides robustness without introducing any material harm.

Error-sensitive locations

New Peak
LiteSEER Heuristic Algorithm

- Heuristic: Perform safety checks only at the corner points of $S$

- Time Complexity
  - $O(n^2)$
  - Lower Bound
Measuring Robustness

- Selectivity Error Resistance Factor (SERF)

\[ SERF(q_e, q_a) = 1 - \frac{c(P_{re}, q_a) - c(P_{oe}, q_a)}{c(P_{oe}, q_a) - c(P_{oa}, q_a)} \]

- At location \( q_a \), fraction of performance gap closed by \( P_{re} \)
Aggregate Impact of Replacements

\[ AggS\text{ERF} = \frac{\sum_{q_e \in rep(S)} \sum_{q_a \in exo_{oe}(S)} S\text{ERF}(q_e, q_a)}{\sum_{q_e \in S} \sum_{q_a \in exo_{oe}(S)} 1} \]

*rep(S)* is the set of query locations in S whose plans were replaced

*exo_{oe}(S)* is the exo-optimal region of \( P_{oe} \) (i.e. set of error locations in S where \( P_{oe} \) is significantly worse than \( P_{oa} \) and robustness is desired)
Performance Metrics

- **AggSERF**: Robustness Metric
- **MaxSERF**: Maximum value of SERF
- **MinSERF**: Minimum value of SERF

- **Rep%**: Percentage of locations where replacement occurred
- **Help%**: Percentage of error instances where replacement reduced the performance gap by at least 2/3
### Robustness Results

#### Overview

- **COST-GREEDY**
- **SEER**
- **LITE-SEER**

<table>
<thead>
<tr>
<th>TPCH QUERY TEMPLATE (TOTAL PLANS)</th>
<th>PLAN</th>
<th>MIN SERF</th>
<th>AGG SERF</th>
<th>REP %</th>
<th>HELP %</th>
<th>PLAN</th>
<th>MIN SERF</th>
<th>AGG SERF</th>
<th>HELP %</th>
<th>PLAN</th>
<th>MIN SERF</th>
<th>AGG SERF</th>
<th>HELP %</th>
</tr>
</thead>
<tbody>
<tr>
<td>QT2 (60)</td>
<td>14</td>
<td>-60.2</td>
<td>-0.04</td>
<td>54</td>
<td>4</td>
<td>7</td>
<td>-0.8</td>
<td>0.18</td>
<td>90</td>
<td>4</td>
<td>-0.8</td>
<td>0.18</td>
<td>90</td>
</tr>
<tr>
<td>QT5 (51)</td>
<td>7</td>
<td>-15.7</td>
<td>0.24</td>
<td>84</td>
<td>32</td>
<td>2</td>
<td>-0.3</td>
<td>0.29</td>
<td>90</td>
<td>24</td>
<td>-0.3</td>
<td>0.29</td>
<td>90</td>
</tr>
<tr>
<td>QT8 (121)</td>
<td>7</td>
<td>-4.5</td>
<td>0.72</td>
<td>89</td>
<td>78</td>
<td>2</td>
<td>-0.3</td>
<td>0.91</td>
<td>99</td>
<td>92</td>
<td>-0.3</td>
<td>0.91</td>
<td>100</td>
</tr>
<tr>
<td>QT9 (137)</td>
<td>9</td>
<td>-33.6</td>
<td>-0.04</td>
<td>86</td>
<td>22</td>
<td>5</td>
<td>-1.4</td>
<td>0.56</td>
<td>99</td>
<td>48</td>
<td>-2.2</td>
<td>0.56</td>
<td>49</td>
</tr>
<tr>
<td>QT10 (44)</td>
<td>3</td>
<td>-24.8</td>
<td>-0.24</td>
<td>85</td>
<td>1</td>
<td>3</td>
<td>-1.0</td>
<td>0.13</td>
<td>98</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Unsafe replacements**

- **Good robustness**
  - + safety
  - + help
  - + anorexia

- **Comparable to SEER**
TUTORIAL OUTLINE

Part I: Plan Diagram Characteristics [VLDB 2005]
Part II: Plan Diagram Production [VLDB 2005/2008]
Part III: Plan Diagram Reduction [VLDB 2007]
Part IV: Robust Plan Diagrams [VLDB 2008]
Part V: Intra-optimizer Integration [VLDB 2010]
Part VI: Future Research Directions
Research Challenge

- SEER/CostGreedy assumed presence of plan diagrams and were “post-facto” solutions for identifying robust replacement plans.
- Can we internalize these ideas in the query optimizer itself such that it online identifies robust plans?
  - i.e. aim for resistance, rather than cure

Fundamental Difficulty: Do not possess global knowledge about behavior in entire selectivity space!
Query Example (~ Q10 of TPCH)

```
select C.custkey, C.name, C.acctbal, N.name
from Customer C, Orders O, Lineitem L, Nation N
where C.custkey = O.custkey and
    L.orderkey = O.orderkey and
    C.nationkey = N.nationkey and
    O.totalprice < 2833 and
    L.extendedprice < 28520
```
Dynamic Programming (DP) Lattice

N – Nation
C – Customer
O – Orders
L – Lineitem

NCOL
322890

NCO
25428

COL
322729

NC
7199

CO
25323

OL
313924

N
1

C
5135

O
16810

L
193584

Cost of the cheapest plan
EXPAND Plan Generation Algorithm
Plan Trains

At error-sensitive nodes of the DP-lattice, form a "plan train" that retains the cheapest plan ("engine") and, in addition, more expensive but stable candidates ("wagons")
Wagon Processing

- **Wagon enumeration**
  - generate candidate set of wagons

- **Wagon pruning**
  - retain only a useful subset
Wagon Enumeration

- Exhaustively “multiply” both input trains
- Costs can be inherited from “engine-engine” multiplication
At each node in lattice, four-stage pruning:

1. Local Cost Check  
   (remove expensive wagons)
2. Global Safety Check  
   (remove unsafe replacements)
3. Global Benefit Check  
   (remove unstable wagons)
4. Cost-Safety-Benefit Skyline Check  
   (remove redundant wagons)
## Wagon Pruning Example [@ NCOL]

<table>
<thead>
<tr>
<th>Local Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>322890</td>
</tr>
<tr>
<td>322901</td>
</tr>
<tr>
<td>324203</td>
</tr>
<tr>
<td>329089</td>
</tr>
<tr>
<td>329100</td>
</tr>
<tr>
<td>329229</td>
</tr>
<tr>
<td>334801</td>
</tr>
<tr>
<td>390748</td>
</tr>
<tr>
<td>395288</td>
</tr>
</tbody>
</table>
Check 1: Local Cost

- Ensure each wagon is near-optimal in absence of errors

- Eliminate all wagon sub-plans $p_w$ with $c(p_w, q_e) > (1 + \lambda) c(p_e, q_e)$
After Local Cost Check ($\lambda = 20\%$)

<table>
<thead>
<tr>
<th>Local Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>322890</td>
</tr>
<tr>
<td>322901</td>
</tr>
<tr>
<td>324203</td>
</tr>
<tr>
<td>329089</td>
</tr>
<tr>
<td>329100</td>
</tr>
<tr>
<td>329229</td>
</tr>
<tr>
<td>334801</td>
</tr>
<tr>
<td>390748</td>
</tr>
<tr>
<td>395288</td>
</tr>
</tbody>
</table>
Check 2: Global Safety

- Wagon $p_w$ is considered safe if it passes the SEER safety test.

- Alternatively, can use the LiteSEER cheap heuristic test for safety:

  $\forall q_a \in \text{corners}(S),
  c(p_w, q_a) \leq (1 + \lambda) c(p_e, q_a)$
After Global Safety Check ($\lambda = 20\%$)

<table>
<thead>
<tr>
<th>Local Cost</th>
<th>$V_0$</th>
<th>$V_1$</th>
<th>$V_2$</th>
<th>$V_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>322890</td>
<td>202089</td>
<td>224599</td>
<td>846630</td>
<td>1271678</td>
</tr>
<tr>
<td>322901</td>
<td>202101</td>
<td>224610</td>
<td>846642</td>
<td>1271689</td>
</tr>
<tr>
<td>324203</td>
<td>202089</td>
<td>224604</td>
<td>846636</td>
<td>1952627</td>
</tr>
<tr>
<td>329089</td>
<td>208207</td>
<td>230766</td>
<td>356555</td>
<td>1280663</td>
</tr>
<tr>
<td>329100</td>
<td>208219</td>
<td>230777</td>
<td>356567</td>
<td>1280674</td>
</tr>
<tr>
<td>329229</td>
<td>202090</td>
<td>224928</td>
<td>846959</td>
<td>4563459</td>
</tr>
<tr>
<td>334801</td>
<td>214078</td>
<td>236628</td>
<td>362417</td>
<td>1204051</td>
</tr>
</tbody>
</table>

$V_i$: Four Corners of $S$
Check 3: Global Benefit

- Benefit Index (heuristic): Arithmetic Mean of corner costs
  \[ \xi(p_w, p_e) = \frac{\overline{c}(p_e, q_a)}{\overline{c}(p_w, q_a)} \quad q_a \in \text{Corners}(S) \]

- Eliminate all \( p_w \) with \( \xi < 1 \)

- Constant ranking property (critical): Same benefit ranking between a given pair of plans at every point in \( S \)
After Global Benefit Check

<table>
<thead>
<tr>
<th>Local Cost</th>
<th>$V_0$</th>
<th>$V_1$</th>
<th>$V_2$</th>
<th>$V_3$</th>
<th>Benefit Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>322890</td>
<td>202089</td>
<td>224599</td>
<td>846630</td>
<td>1271678</td>
<td>1.0</td>
</tr>
<tr>
<td>322901</td>
<td>202101</td>
<td>224610</td>
<td>846642</td>
<td>1271689</td>
<td>0.99</td>
</tr>
<tr>
<td>329089</td>
<td>208207</td>
<td>230766</td>
<td>356555</td>
<td>1280663</td>
<td>1.22</td>
</tr>
<tr>
<td>329100</td>
<td>208219</td>
<td>230777</td>
<td>356567</td>
<td>1280674</td>
<td>1.22</td>
</tr>
<tr>
<td>334801</td>
<td>214078</td>
<td>236628</td>
<td>362417</td>
<td>1204051</td>
<td>1.26</td>
</tr>
</tbody>
</table>
Check 4: Cost-Safety-Benefit Skyline

- Eliminates “dominated” wagons
- Corner costs \((V_0, V_1, V_2, V_3)\) form the skyline dimensions
  - Benefit dimension implied with Arithmetic Mean
- Skyline set of wagons is equivalent to retaining the entire set of wagons
[proof in paper]
### After CSB Skyline Check

<table>
<thead>
<tr>
<th>Local Cost</th>
<th>$V_0$</th>
<th>$V_1$</th>
<th>$V_2$</th>
<th>$V_3$</th>
<th>Benefit Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>322890</td>
<td>202089</td>
<td>224599</td>
<td>846630</td>
<td>1271678</td>
<td>1.0</td>
</tr>
<tr>
<td>329089</td>
<td>208207</td>
<td>230766</td>
<td>356555</td>
<td>1280663</td>
<td>1.22</td>
</tr>
<tr>
<td>329100</td>
<td>208219</td>
<td>230777</td>
<td>356567</td>
<td>1280674</td>
<td>1.22</td>
</tr>
<tr>
<td>334801</td>
<td>214078</td>
<td>236628</td>
<td>362417</td>
<td>1204051</td>
<td>1.26</td>
</tr>
</tbody>
</table>
Final Plan Selection

- If internal node, forward the entire train to upper lattice nodes
- If root node, pick the complete plan with the greatest benefit index.
  - could be the engine itself or a wagon

<table>
<thead>
<tr>
<th>Local Cost</th>
<th>$V_0$</th>
<th>$V_1$</th>
<th>$V_2$</th>
<th>$V_3$</th>
<th>Benefit Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>322890</td>
<td>202089</td>
<td>224599</td>
<td>846630</td>
<td>1271678</td>
<td>1.0329089</td>
</tr>
<tr>
<td>329089</td>
<td>208207</td>
<td>230766</td>
<td>356555</td>
<td>1280663</td>
<td>1.22</td>
</tr>
<tr>
<td>334801</td>
<td>214078</td>
<td>236628</td>
<td>362417</td>
<td>1204051</td>
<td>1.26</td>
</tr>
</tbody>
</table>

Big difference!
Implementation

- Query Optimizer: PostgreSQL 8.3.6
- Implemented Foreign Plan Costing
  - Complication due to PostgreSQL cacheing certain temporary results during the optimization process which have an impact on the final plan costs
- Optimization objective solely response-time, not a combination of response-time and latency
- About 10K lines of code, mostly for FPC
  - easy to extend to other optimizers
1. Plan Robustness Performance

- Performance comparable to SEER (global knowledge)!

### Table: Expanded SEER

<table>
<thead>
<tr>
<th>Query</th>
<th>TEMP</th>
<th>REP%</th>
<th>Agg SERF</th>
<th>Max</th>
<th>Help</th>
<th>R EP</th>
<th>Agg SERF</th>
<th>Max</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>QT5</td>
<td></td>
<td>85</td>
<td>0.54</td>
<td>1</td>
<td>55</td>
<td>47</td>
<td>0.61</td>
<td>1</td>
<td>64</td>
</tr>
<tr>
<td>QT10</td>
<td></td>
<td>98</td>
<td>0.21</td>
<td>1</td>
<td>20</td>
<td>37</td>
<td>0.21</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>3DQT8</td>
<td></td>
<td>69</td>
<td>0.18</td>
<td>1</td>
<td>10</td>
<td>59</td>
<td>0.17</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>3DQT10</td>
<td></td>
<td>66</td>
<td>0.39</td>
<td>1</td>
<td>44</td>
<td>24</td>
<td>0.38</td>
<td>1</td>
<td>41</td>
</tr>
<tr>
<td>DSQT7</td>
<td></td>
<td>93</td>
<td>0.28</td>
<td>1</td>
<td>28</td>
<td>46</td>
<td>0.28</td>
<td>1</td>
<td>28</td>
</tr>
<tr>
<td>DSQT26</td>
<td></td>
<td>30</td>
<td>0.49</td>
<td>1</td>
<td>50</td>
<td>29</td>
<td>0.49</td>
<td>1</td>
<td>49</td>
</tr>
</tbody>
</table>

- High replacement %
- Error Immunity
- Good Robustness
- Good Help%
## 2. Plan Diagram Characteristics

### Query Template Plans

<table>
<thead>
<tr>
<th>Query Template</th>
<th>POP Plans</th>
<th>EXPAND Plans</th>
<th>SEER Plans</th>
</tr>
</thead>
<tbody>
<tr>
<td>QT5</td>
<td>11</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>QT10</td>
<td>15</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>3DQT8</td>
<td>43</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>3DQT10</td>
<td>30</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>DSQT7</td>
<td>12</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>DSQT26</td>
<td>13</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

- **Anorexic diagrams**
- **Non-POSP plans**
Sample Plan Diagrams

[April 2011 Plan Diagrams Tutorial (ICDE 2011)]

DP: 28 plans

EXPAND: 3 plans

Non-POSP plan
## 3. Time Overheads

<table>
<thead>
<tr>
<th>Query Template</th>
<th>Optimization Time (ms)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DP</td>
<td>EXPAND</td>
</tr>
<tr>
<td>QT5</td>
<td>3.2</td>
<td>22.2 (+19.0)</td>
</tr>
<tr>
<td>QT10</td>
<td>0.9</td>
<td>3.2 (+2.3)</td>
</tr>
<tr>
<td>3DQT8</td>
<td>3.5</td>
<td>30.6 (+27.1)</td>
</tr>
<tr>
<td>3DQT10</td>
<td>0.9</td>
<td>4.3 (+3.4)</td>
</tr>
<tr>
<td>DSQT7</td>
<td>1.3</td>
<td>7.7 (+6.4)</td>
</tr>
<tr>
<td>DSQT26</td>
<td>1.4</td>
<td>7.0 (+5.6)</td>
</tr>
</tbody>
</table>

- Additional time of < 100ms
  - Miniscule compared to expected execution time savings
4. Memory Overheads

<table>
<thead>
<tr>
<th>Query Template</th>
<th>Memory Overheads (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DP</td>
</tr>
<tr>
<td>QT5</td>
<td>2.8</td>
</tr>
<tr>
<td>QT10</td>
<td>2.2</td>
</tr>
<tr>
<td>3DQT8</td>
<td>4.0</td>
</tr>
<tr>
<td>3DQT10</td>
<td>2.2</td>
</tr>
<tr>
<td>DSQT7</td>
<td>2.4</td>
</tr>
<tr>
<td>DSQT26</td>
<td>2.4</td>
</tr>
</tbody>
</table>

- Extra memory of < 100 MB
- Held very briefly (< 100 ms)
EXPAND is an effective all-round choice for incorporation in industrial-strength database query optimizers, delivering online computation, good plan robustness, replacement safety, anorexic plan diagrams, and acceptable overheads.
Take Away

- Dense and Intricate Plan Diagrams
- PQO violations
- Optimizer bugs

Cost Greedy

- Anorexic Reduction (less than 10 plans)
- Local Near-optimality (20%)

SEER / GSPQO

- Anorexic Reduction
- Global Safety (“no harm”)
- Robust Plans
- Efficient Approximation

EXPAND

- Anorexic Reduction
- Global Safety
- Robust Plans
- Online Processing

[VLDB05]
[VLDB07]
[VLDB08]
[VLDB10]
TUTORIAL OUTLINE

Part I: Plan Diagram Characteristics [VLDB 2005]
Part II: Plan Diagram Production [VLDB 2005/2008]
Part III: Plan Diagram Reduction [VLDB 2007]
Part IV: Robust Plan Diagrams [VLDB 2008]
Part V: Intra-optimizer Integration [VLDB 2010]
Part VI: Future Research Directions
1. Diagram Density Classifier

Develop a quantitative predictor for diagram density prior to production

Data mining problem with feature vector including aspects of the query graph, database optimizer, and database statistics.
2. Diagram Coloring Scheme

Assign plan colors based on **structural differences**. For instance, if a pair of plans have same join order, assign close shades of a common color.

Plan diagram itself provides a reflection of the differences between plans in the selectivity space.

To achieve this objective, a semantically consistent **plan distance metric** needs to be defined, after which an efficient coloring scheme that closely reflects these differences has to be designed.
We have empirically shown the anorexic nature of plan diagram reduction. It would be interesting to assess whether a formal theory could be established that explains the observed behavior.
4. Fully Robust Plans

EXPAND/SEER schemes provide robustness to selectivity estimation errors on base relation selection predicates.

Extend to achieve robustness to selectivity estimation errors anywhere in the plan tree (e.g. join selectivity errors).

Would result in “bulletproof” complete query execution plans.
Plan diagrams capture the “compile-time” behavior of query optimizers. Useful to also visualize the run-time behavior in a similar manner [CIDR2009]
“Analyzing Plan Diagrams of Database Query Optimizers”
*Proc. of 31st Intl. Conf. on Very Large Data Bases (VLDB)*, 2005.

“On the Production of Anorexic Plan Diagrams”
*Proc. of 33th Intl. Conf. on Very Large Data Bases (VLDB)*, 2007.

“Identifying Robust Plans through Plan Diagram Reduction”
*Proc. of 34th Intl. Conf. on Very Large Data Bases (VLDB)*, 2008.

“Efficiently Approximating Query Optimizer Plan Diagrams”
*Proc. of 34th Intl. Conf. on Very Large Data Bases (VLDB)*, 2008.

“On the Stability of Plan Costs and the Costs of Plan Stability”
*Proc. of 36th Intl. Conf. on Very Large Data Bases (VLDB)*, 2010.

“The Picasso Database Query Optimizer Visualizer”
*Proc. of 36th Intl. Conf. on Very Large Data Bases (VLDB)*, 2010.
http://dsl.serc.iisc.ernet.in/projects/PICASSO

Publications, Software, Sample Diagrams
Acknowledgements

- **Sponsors:** Dept. of Science and Technology, Govt. of India, Google, IBM, Microsoft, Sybase

- **Students:** Dedicated efforts of a large group of students, listed on the Picasso website