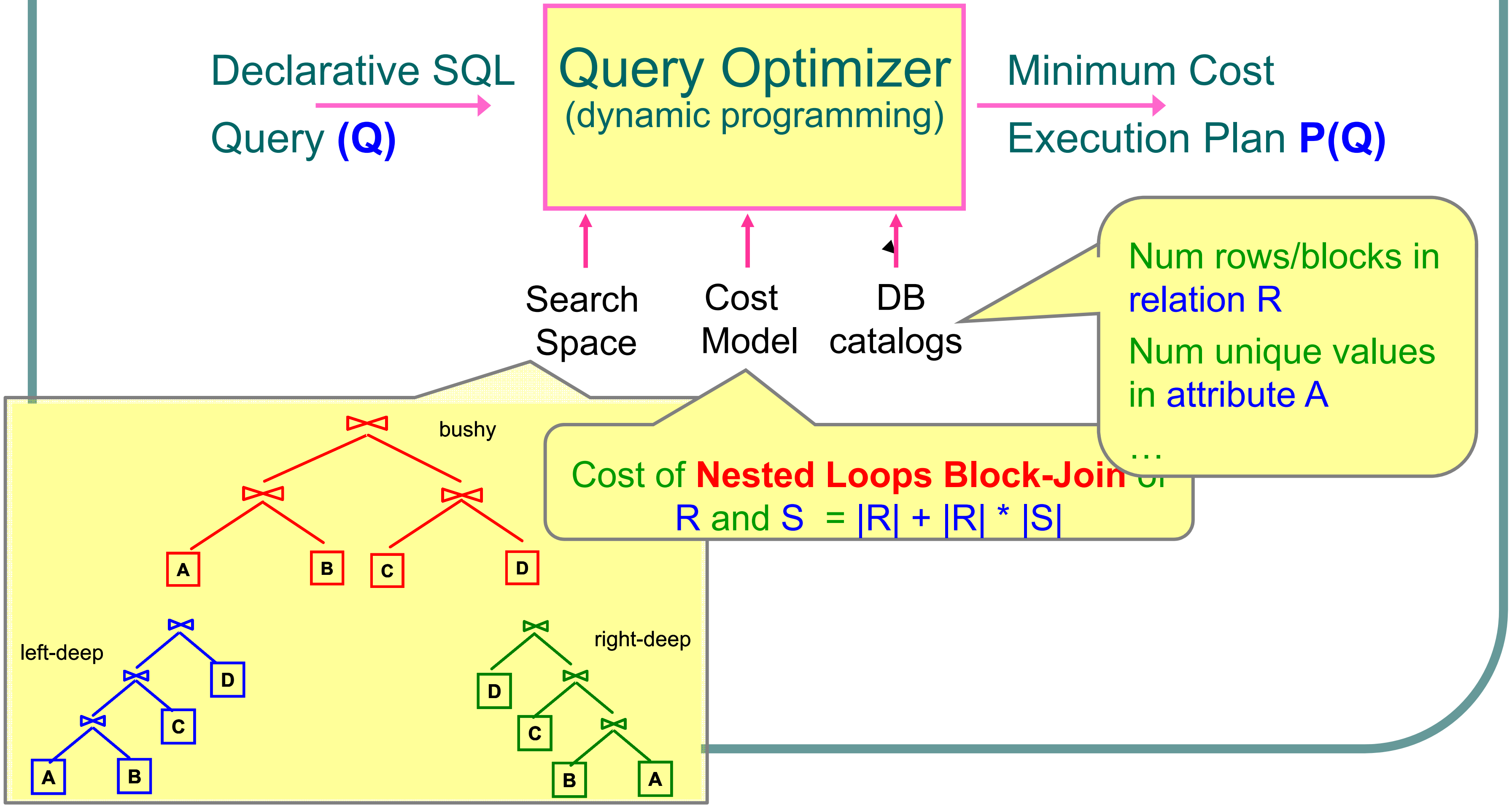
Query Optimizer Plan Diagrams: Production, Reduction and Applications

Jayant Haritsa

Database Systems Lab
Indian Institute of Science
Bangalore, INDIA

Cost-based Query Optimization



Declarative SQL Query **(Q)**

Query Optimizer (dynamic programming)

Minimum Cost Execution Plan **P(Q)**

Search Space

Cost Model

DB catalogs

Num rows/blocks in relation R

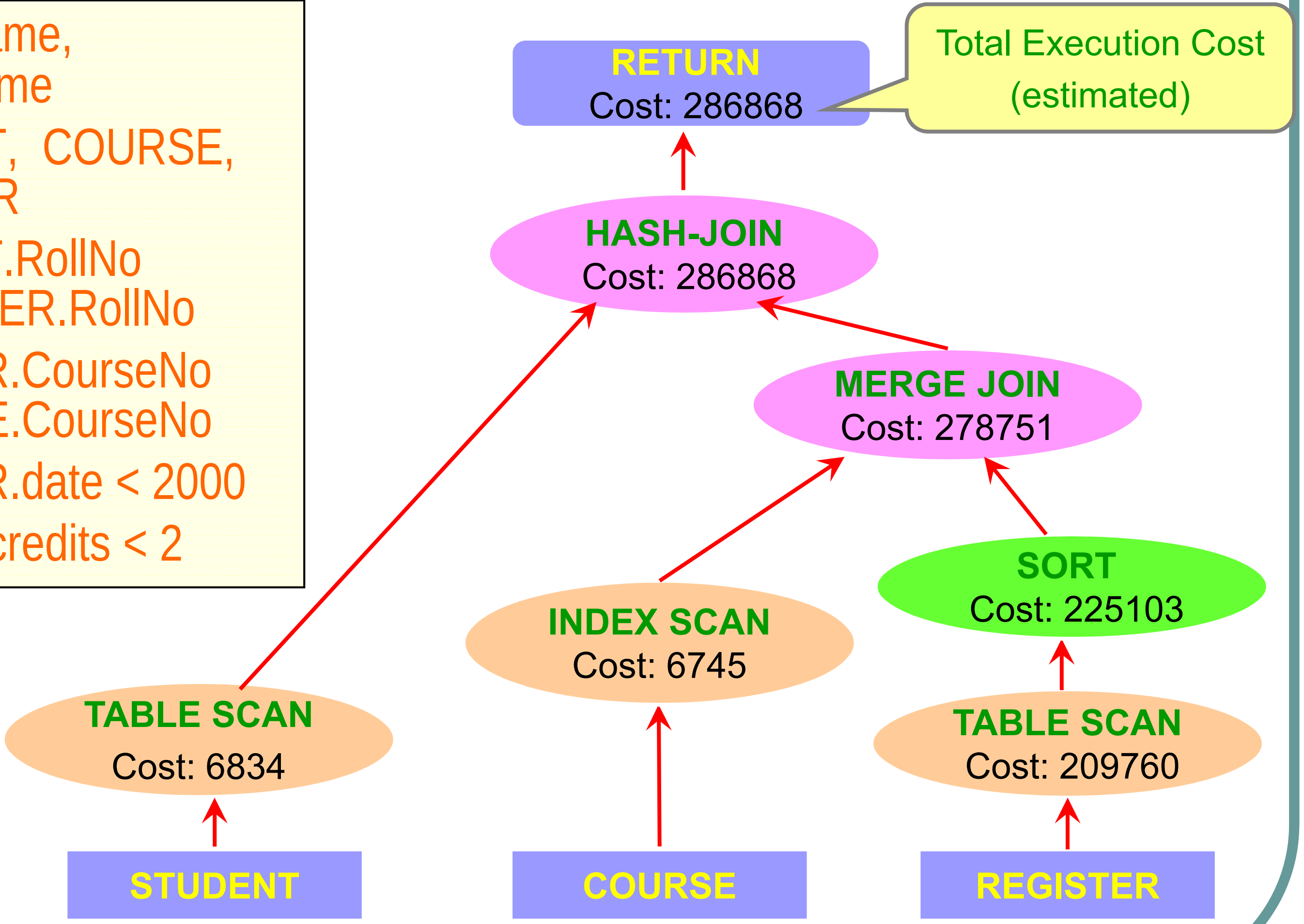
Num unique values in attribute A

…

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

Plan Example

```
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)

HASH-JOIN
Cost: 286868

MERGE JOIN
Cost: 278751

SORT
Cost: 225103

INDEX SCAN
Cost: 6745

TABLE SCAN
Cost: 6834

TABLE SCAN
Cost: 209760

STUDENT

COURSE

REGISTER

Relational Selectivities

- Cost-based Query Optimizer's choice of

  execution plan = f (query, database, system, …)

- For a given database and system setup,

  execution plan chosen for a query =
  f (selectivities of query's base relations)

  - selectivity is the estimated percentage of rows of a relation used in producing the query result

Query Template QT7
[Q7 of TPC-H]

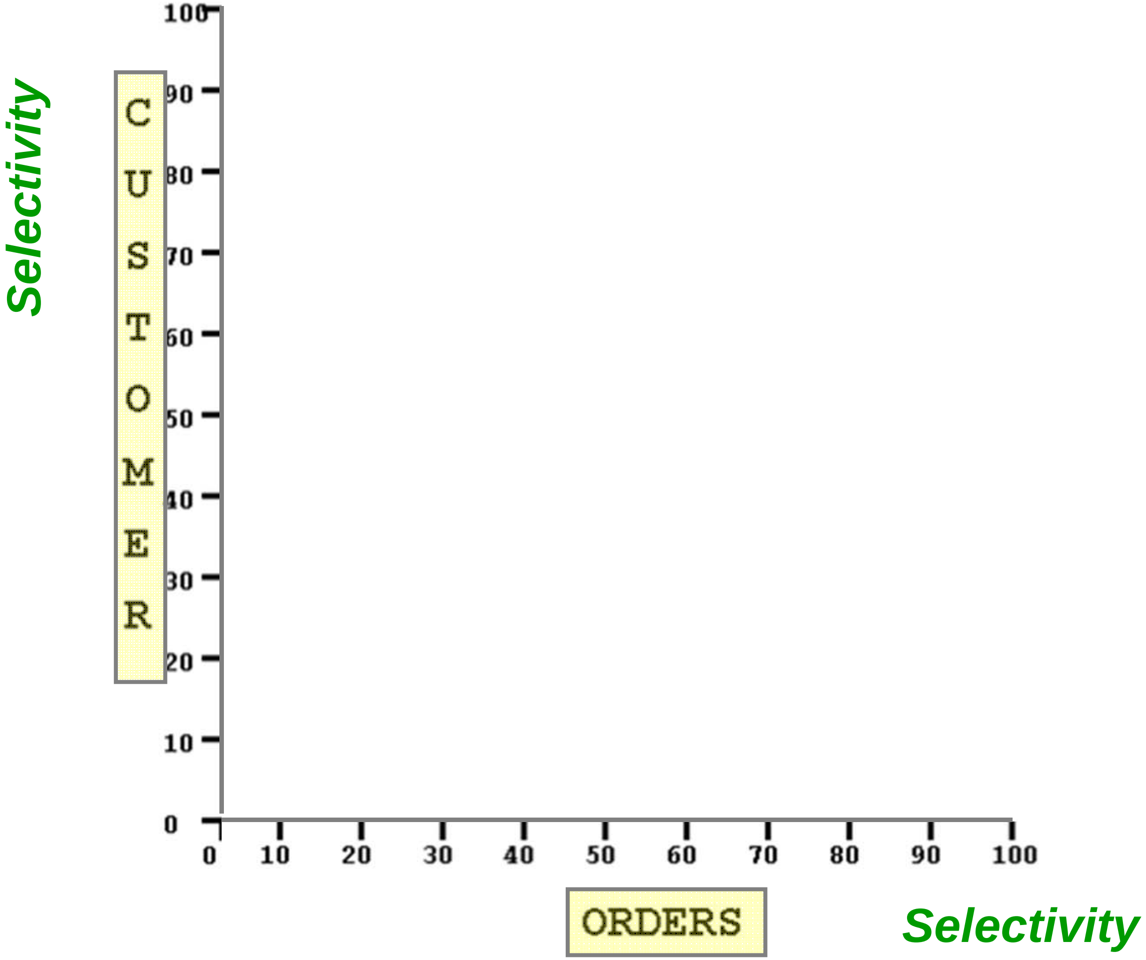
Determines the values of goods shipped between nations in a time period

```
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
 wher                              okey and o_orderkey = l
                                   key and s_nationkey = n
                                   n2.n_nationkey and
        ((n1.n_name        RANCE' and n2.n_name = 'GE       ANY') or
         (n1.n_name = 'G    RMANY' and n2.n_name = '     ANCE')) and
        l_shipdate between  date '1995-01-01' and da    '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
```

Value determines selectivity of ORDERS relation

Value determines selectivity of CUSTOMER relation

Relational Selectivity Space



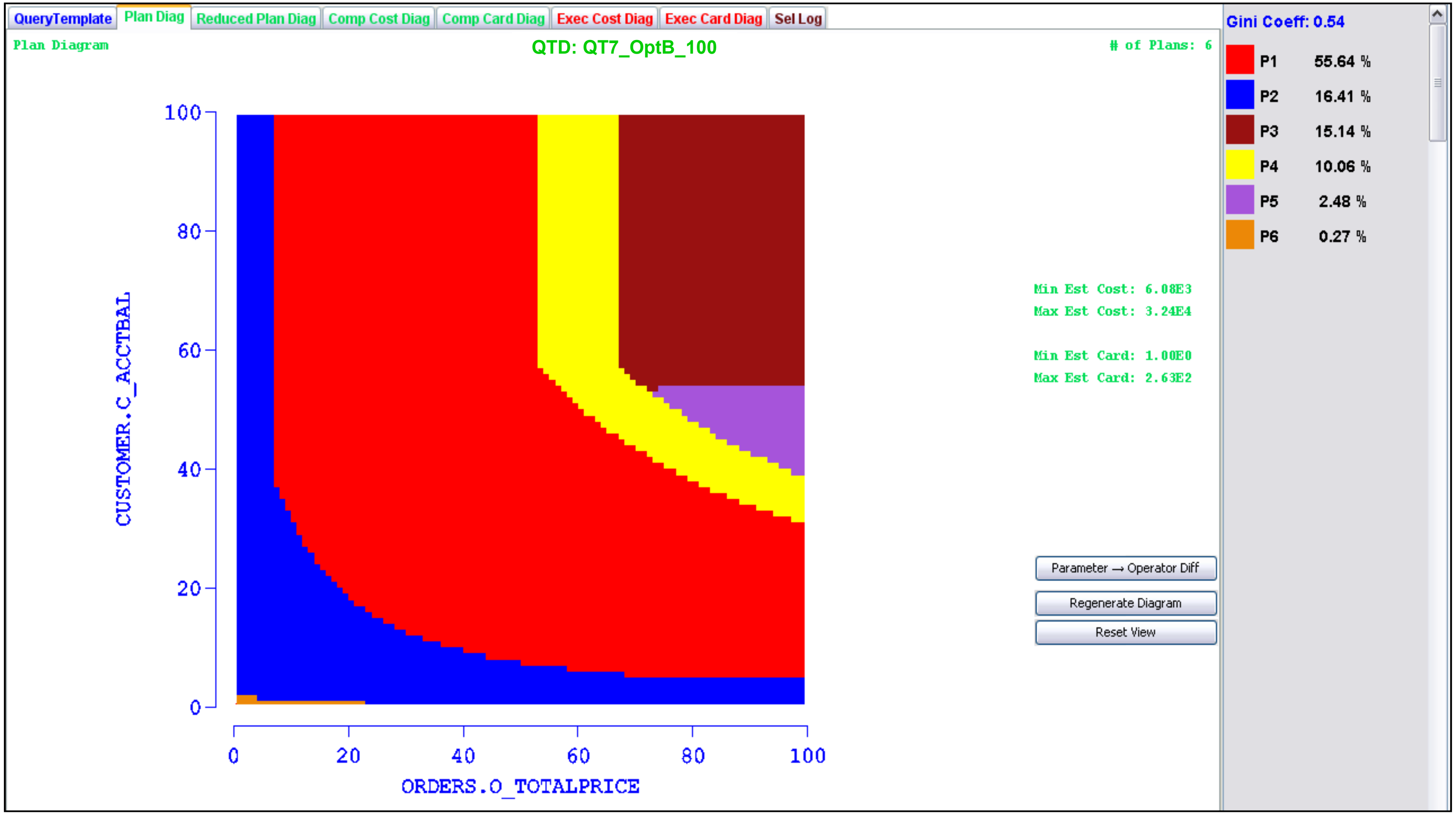
Plan and Cost Diagrams

- 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]



Plan Diagram

QTD: QT7_OptB_100

# of Plans: 6

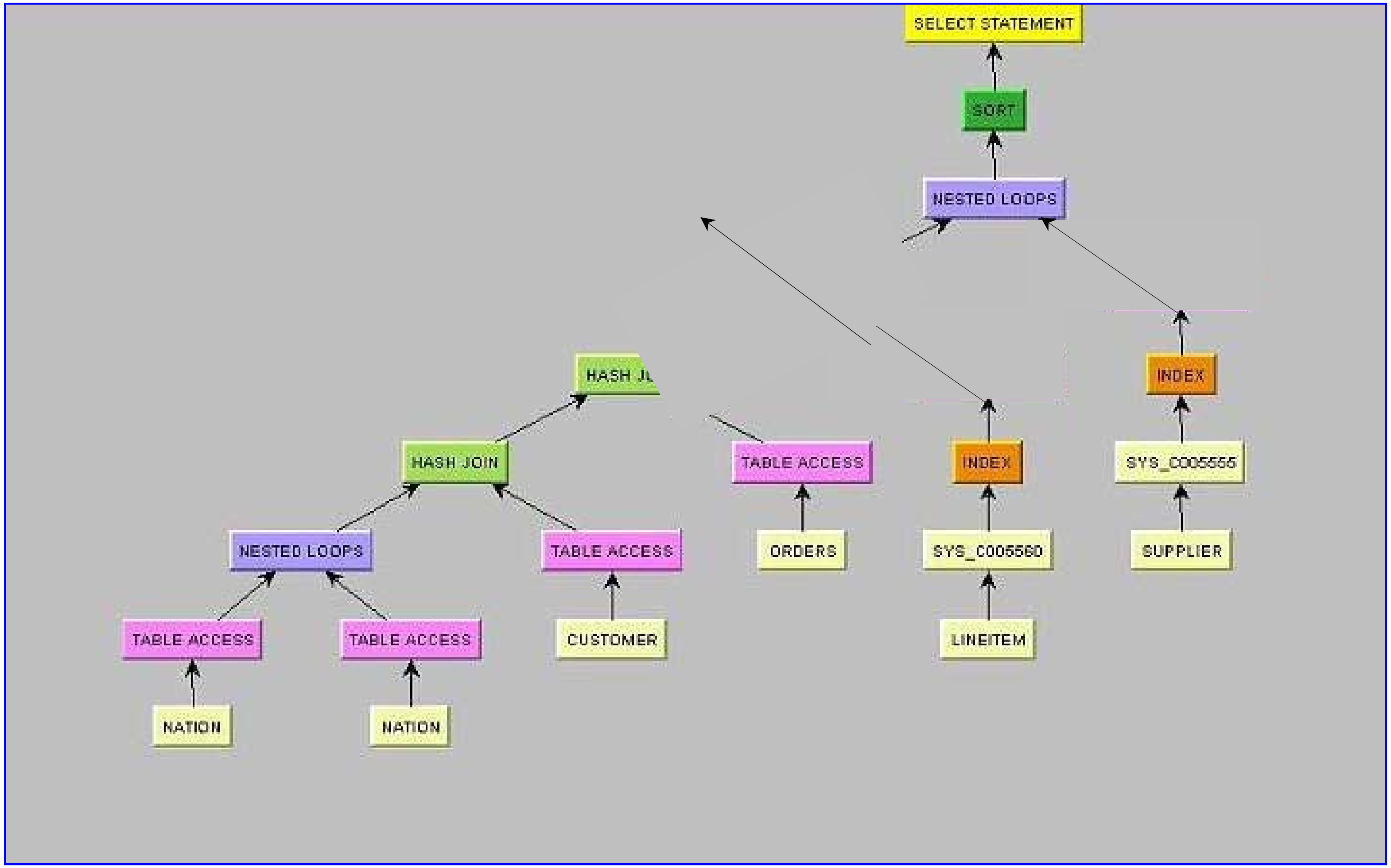
Min Est Cost: 6.08E3
Max Est Cost: 3.24E4

Min Est Card: 1.00E0
Max Est Card: 2.63E2

Gini Coeff: 0.54

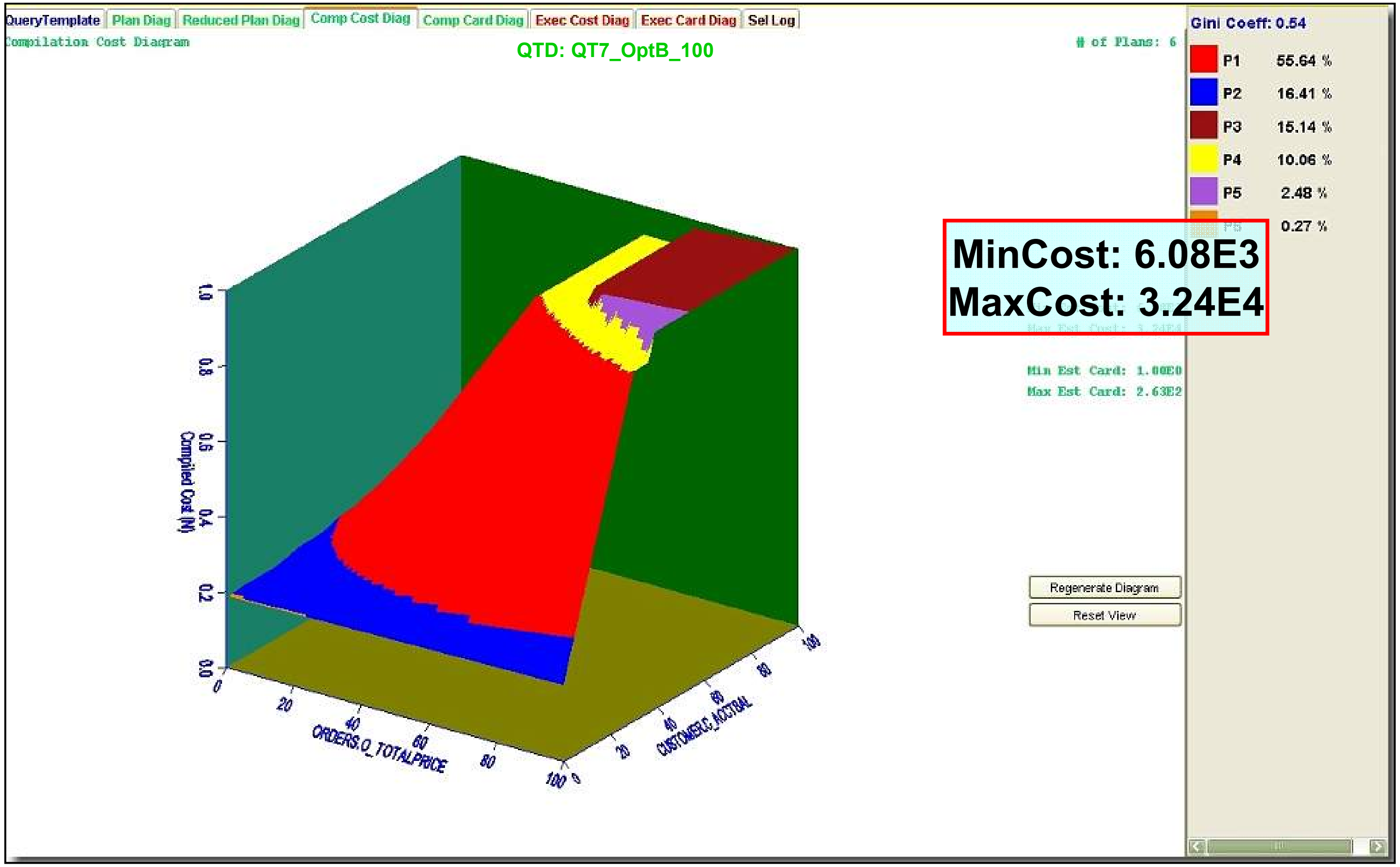
🟥	P1	55.64 %
🟦	P2	16.41 %
🟫	P3	15.14 %
🟨	P4	10.06 %
🟪	P5	2.48 %
🟧	P6	0.27 %

Plan P1


SELECT STATEMENT
SORT
NESTED LOOPS
HASH J
INDEX
HASH JOIN
TABLE ACCESS
INDEX
SYS_C005555
NESTED LOOPS
TABLE ACCESS
ORDERS
SYS_C005560
SUPPLIER
TABLE ACCESS
TABLE ACCESS
CUSTOMER
LINEITEM
NATION
NATION

Sample Cost Diagram

[QT7,OptB, Res=100]



Compilation Cost Diagram

QTD: QT7_OptB_100

# of Plans: 6

Gini Coeff: 0.54

🟥	P1	55.64 %
🟦	P2	16.41 %
🟫	P3	15.14 %
🟨	P4	10.06 %
🟪	P5	2.48 %
🟧	P6	0.27 %

MinCost: 6.08E3
MaxCost: 3.24E4

Min Est Card: 1.00E0
Max Est Card: 2.63E2

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

TPC-H Relation	Relation Cardinality
REGION	5
NATION	25
SUPPLIER	10000
CUSTOMER	150000
PART	200000
PARTSUPP	800000
ORDERS	1500000
LINEITEM	6001215

- 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

The Picasso Connection

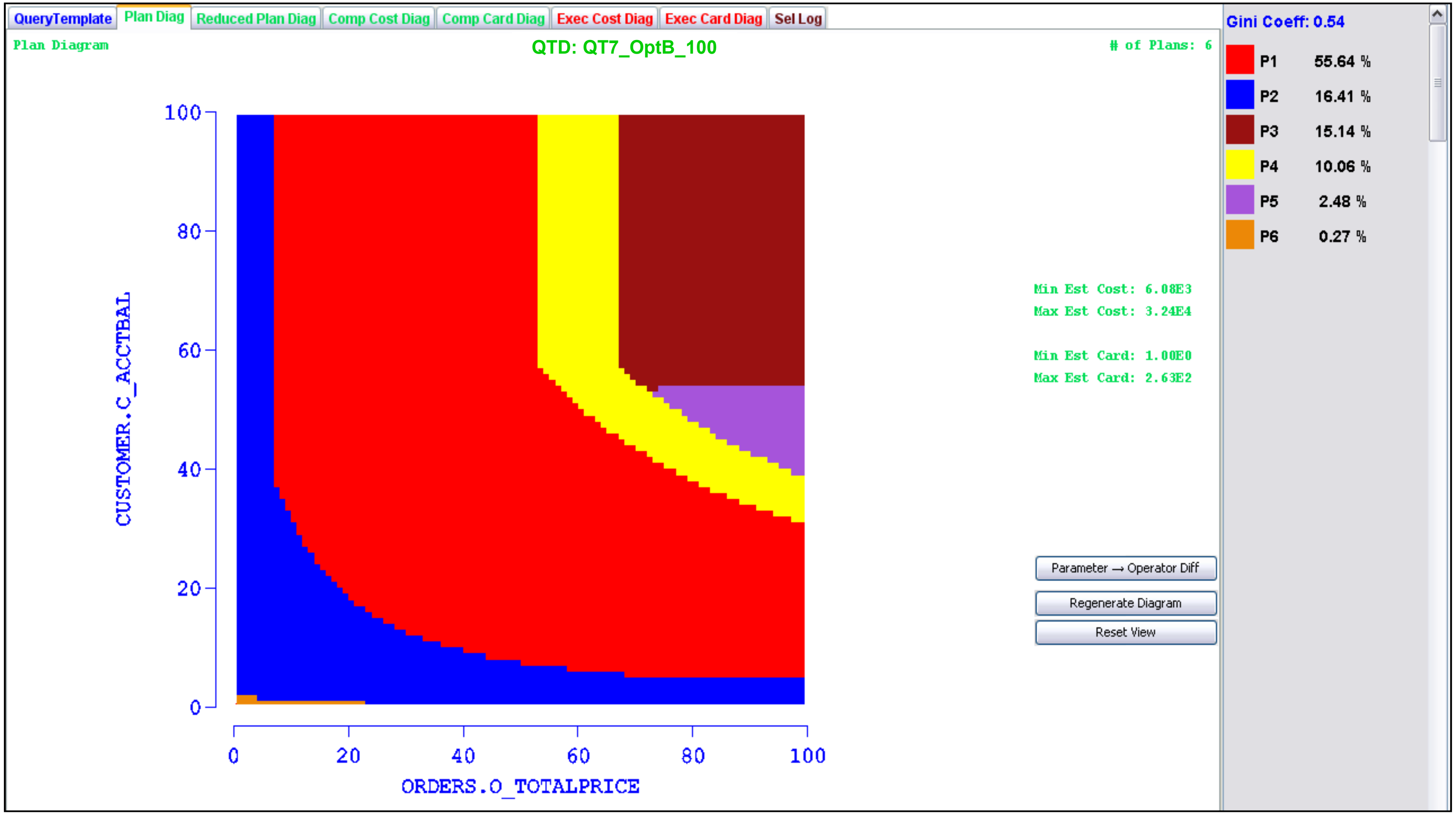
Woman with a guitar
Georges Braque, 1913

Plan diagrams are often similar to cubist paintings !

[Pablo Picasso –
founder of cubist genre]

Smooth Plan Diagram

[QT7, OptB, Res=100]



QTD: QT7_OptB_100

# of Plans: 6

Gini Coeff: 0.54

🟥	P1	55.64 %
🟦	P2	16.41 %
🟫	P3	15.14 %
🟨	P4	10.06 %
🟪	P5	2.48 %
🟧	P6	0.27 %

Min Est Cost: 6.08E3
Max Est Cost: 3.24E4

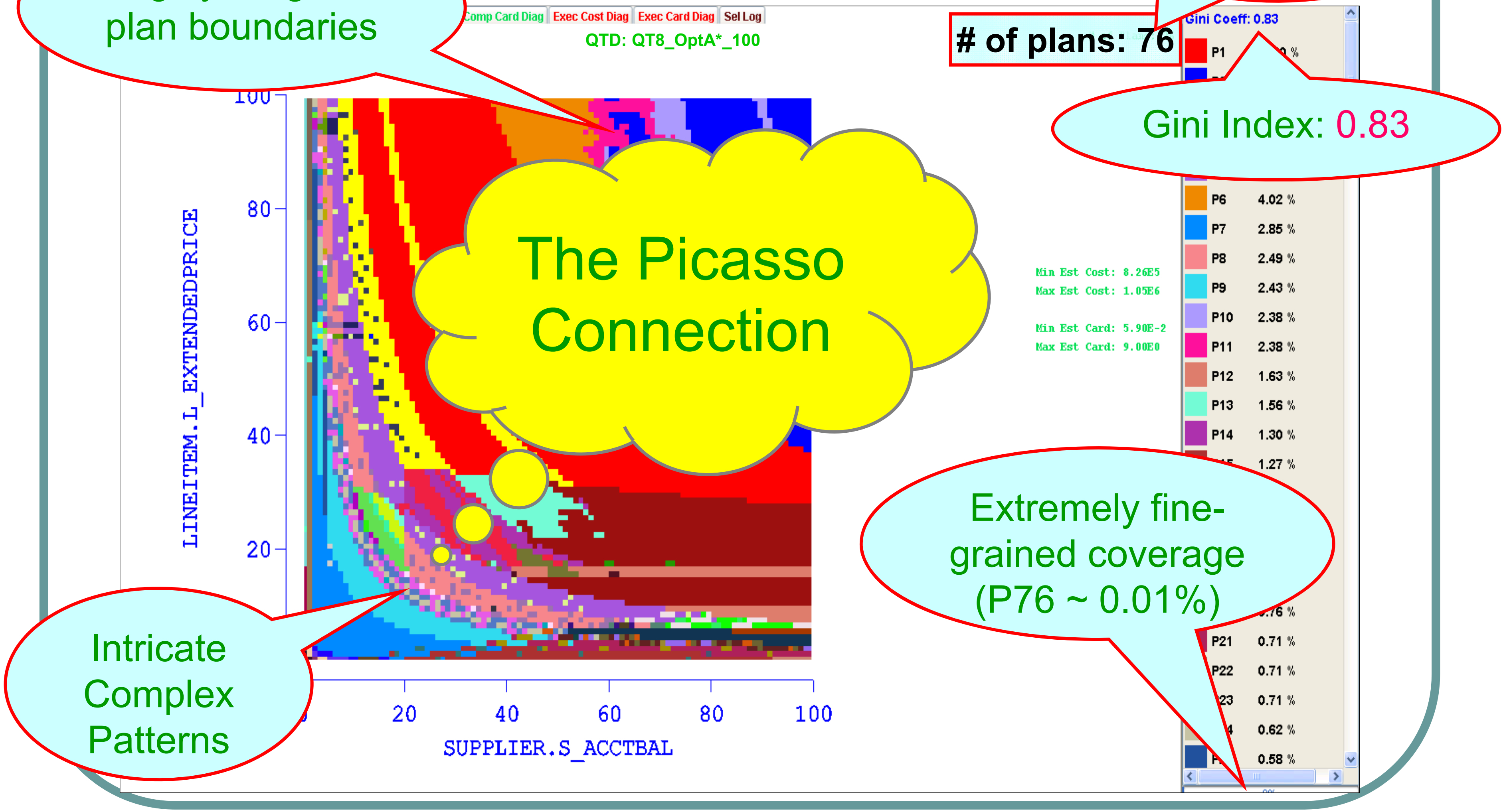
Min Est Card: 1.00E0
Max Est Card: 2.63E2

Complex Plan Diagram

[QT8,OptA*_Res=100]

Increases to **90 plans** with 300x300 grid !

Highly irregular plan boundaries



QTD: QT8_OptA*_100

# of plans: 76

Gini Coeff: 0.83

Gini Index: 0.83

The Picasso Connection

Min Est Cost: 8.26E5
Max Est Cost: 1.05E6

Min Est Card: 5.90E-2
Max Est Card: 9.00E0

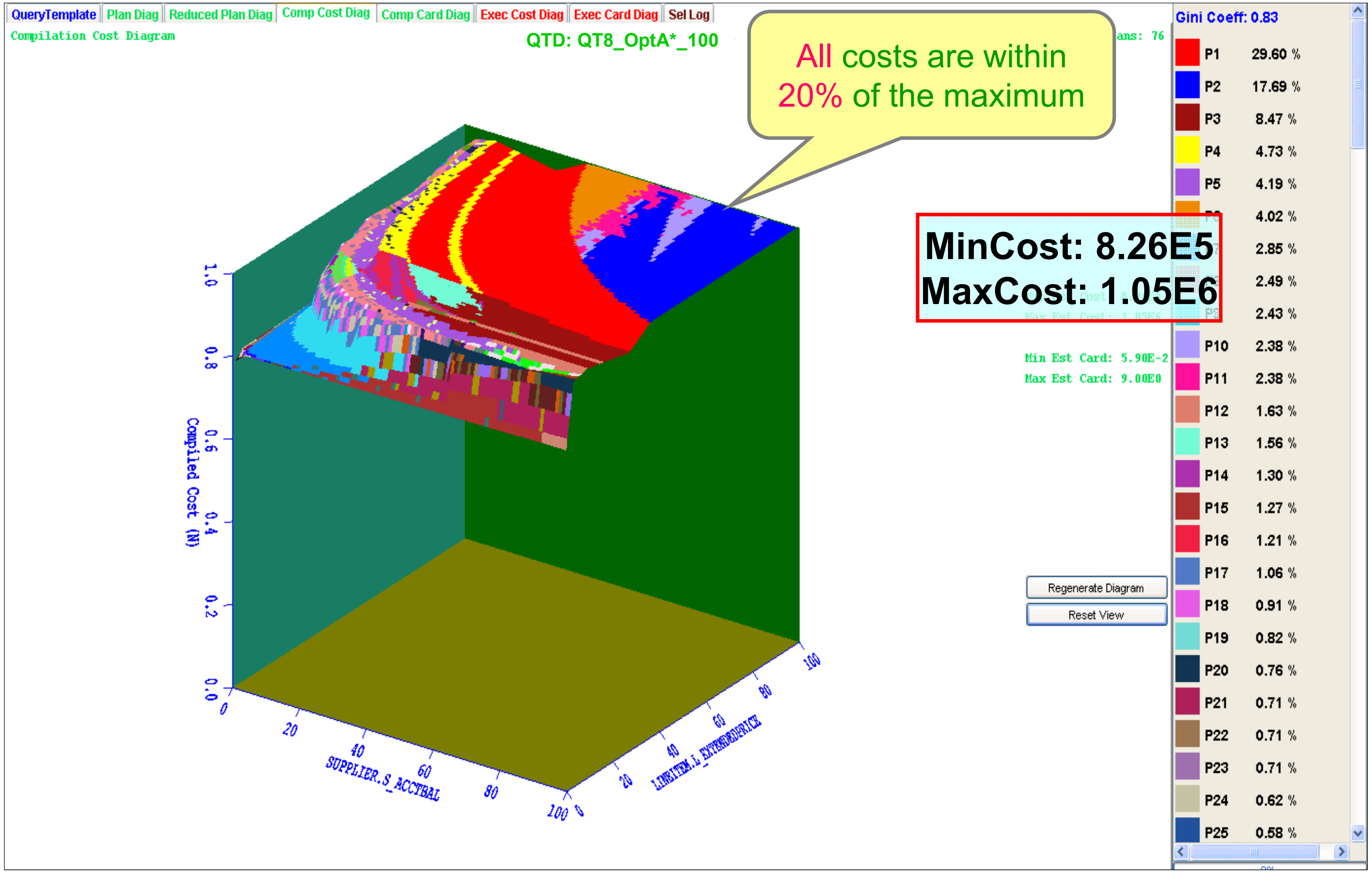
🟥	P1	
🟧	P6	4.02 %
🟦	P7	2.85 %
	P8	2.49 %
	P9	2.43 %
	P10	2.38 %
	P11	2.38 %
	P12	1.63 %
	P13	1.56 %
🟪	P14	1.30 %
		1.27 %
		.76 %
	P21	0.71 %
	P22	0.71 %
	23	0.71 %
	4	0.62 %
		0.58 %

Extremely fine-grained coverage (P76 ~ 0.01%)

Intricate Complex Patterns

Cost Diagram

[QT8, Opt A*, Res=100]



All costs are within 20% of the maximum

MinCost: 8.26E5
MaxCost: 1.05E6

Min Est Card: 5.90E-2
Max Est Card: 9.00E0

Gini Coeff: 0.83

🟥	P1	29.60 %
🟦	P2	17.69 %
🟫	P3	8.47 %
🟨	P4	4.73 %
🟪	P5	4.19 %
🟧		4.02 %
		2.85 %
		2.49 %
		2.43 %
🟪	P10	2.38 %
🟥	P11	2.38 %
🟫	P12	1.63 %
🟩	P13	1.56 %
🟪	P14	1.30 %
🟫	P15	1.27 %
🟥	P16	1.21 %
🟦	P17	1.06 %
🟪	P18	0.91 %
🟩	P19	0.82 %
⬛	P20	0.76 %
🟥	P21	0.71 %
🟫	P22	0.71 %
🟪	P23	0.71 %
⬜	P24	0.62 %
🟦	P25	0.58 %

Plan Space Coverage

[Res=100]

80-20 Rule
Gini skew index > 0.5

TPCH Query Template	Opt A			Opt B			Opt C		
	Plan Cardinality	80% Coverage	Gini Index	Plan Cardinality	80% Coverage	Gini Index	Plan Cardinality	80% Coverage	Gini Index
QT2									
QT5									
QT7									
QT8	**31**			**25**			**38**		
QT9	**63**			**44**			**41**		
QT10									
QT18	**5**			**13**			**5**		
QT21									
Average									

Picasso Art Gallery

- Duplicates and Islands
- Plan Switch Points
- Venetian Blinds
- Footprint Pattern
- Speckle Pattern

Duplicates and

[QT10, Op

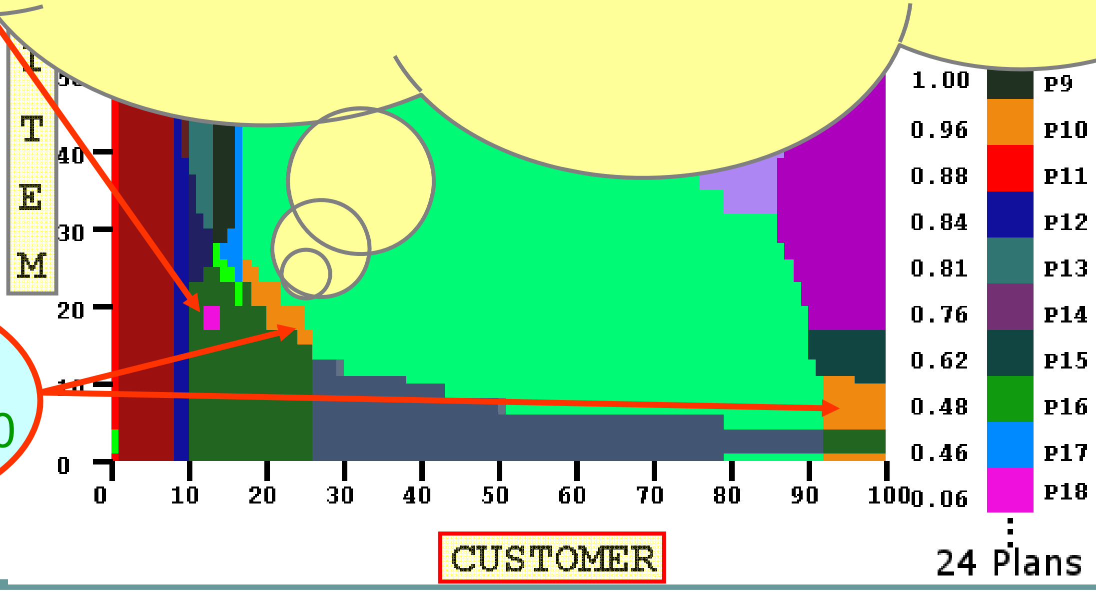
Violates basic tenets of Parametric Query Optimization (PQO) literature:

- 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.

P18

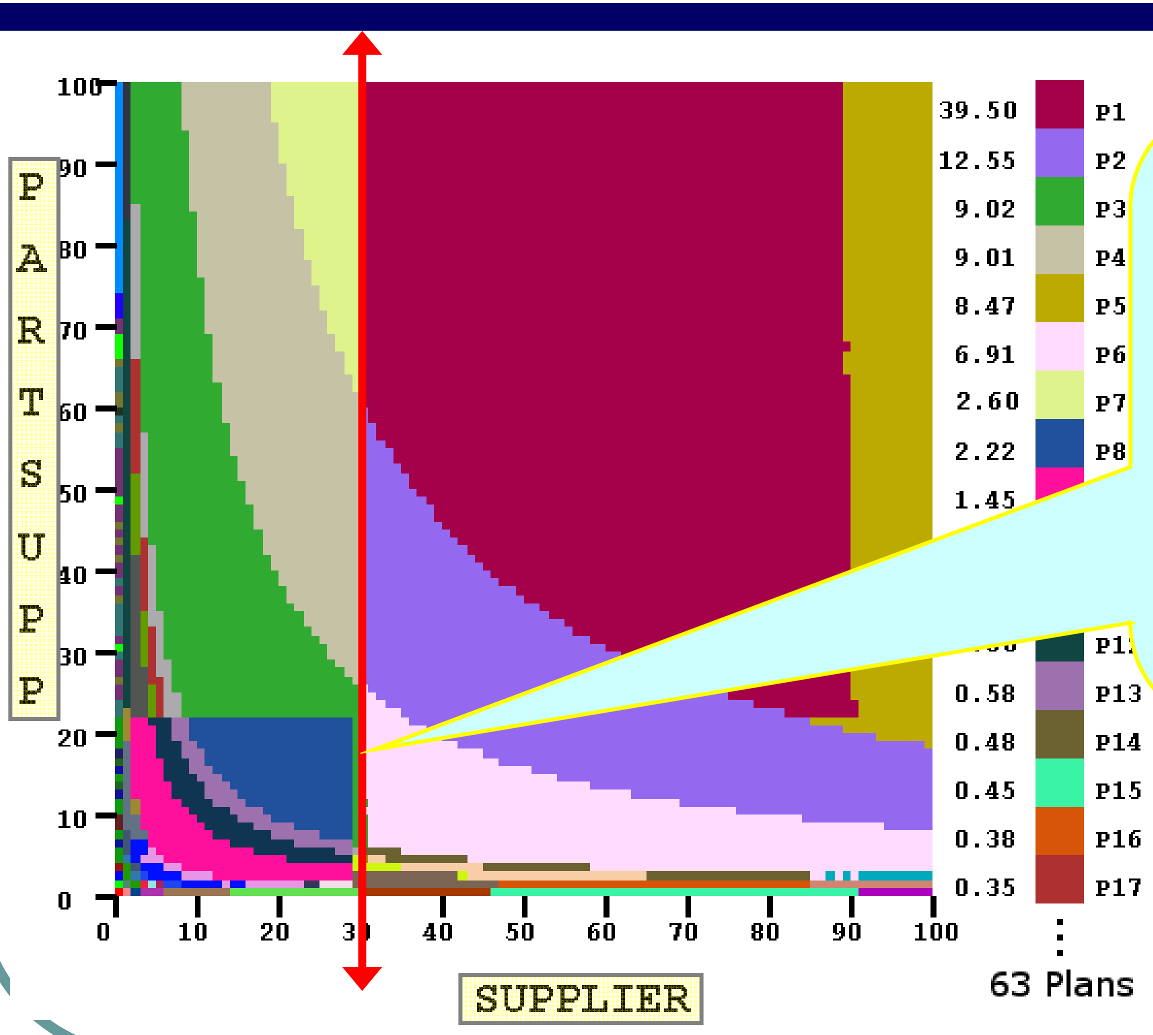
island w

Duplicate locations of P10



Plan Switch Points

[QT9,OptA]

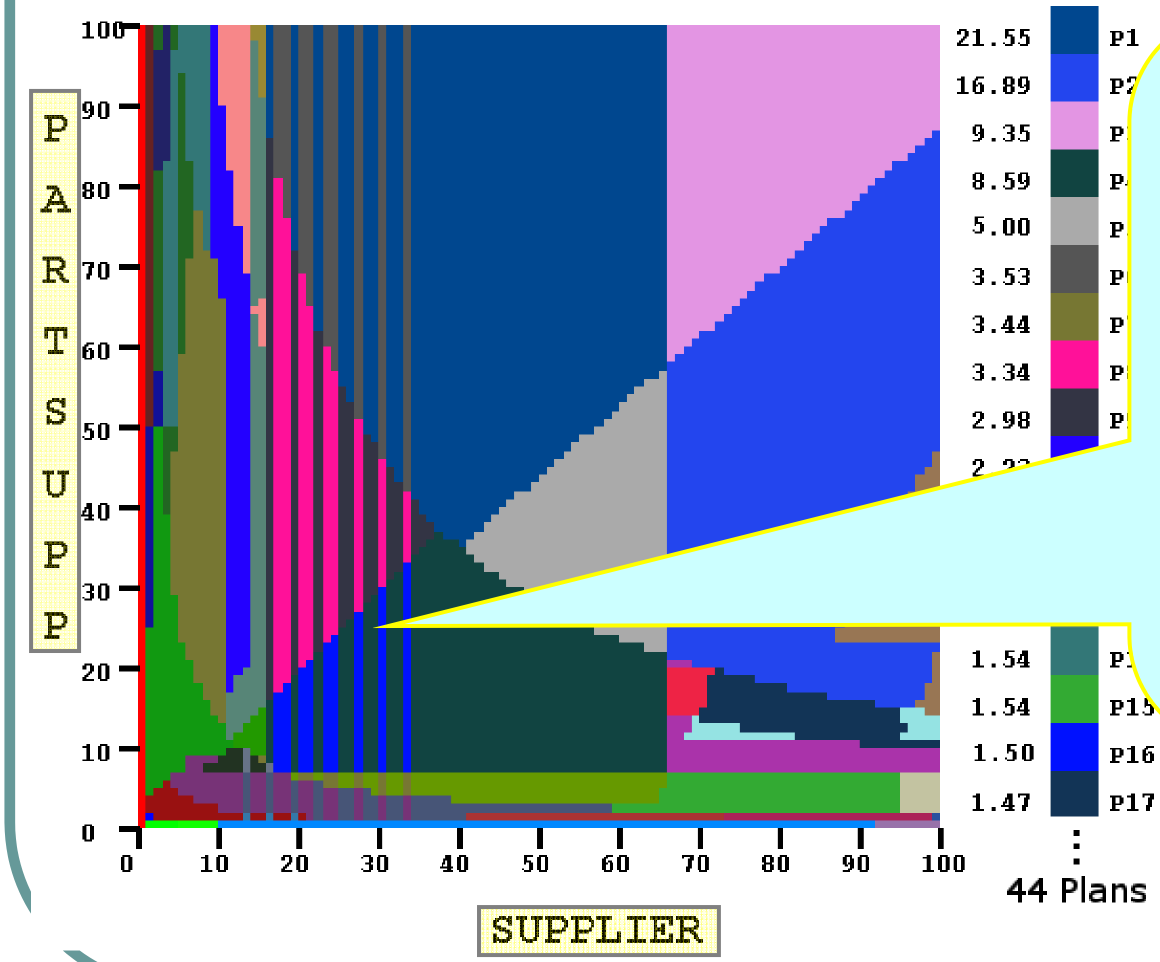


Plan Switch Point:
line parallel to axis with a plan shift for all plans bordering the line.

Hash-Join sequence
PARTSUPP ⋈ SUPPLIER ⋈ PART
is altered to
PARTSUPP ⋈ PART ⋈ SUPPLIER

Venetian Blinds

[QT9,OptB]

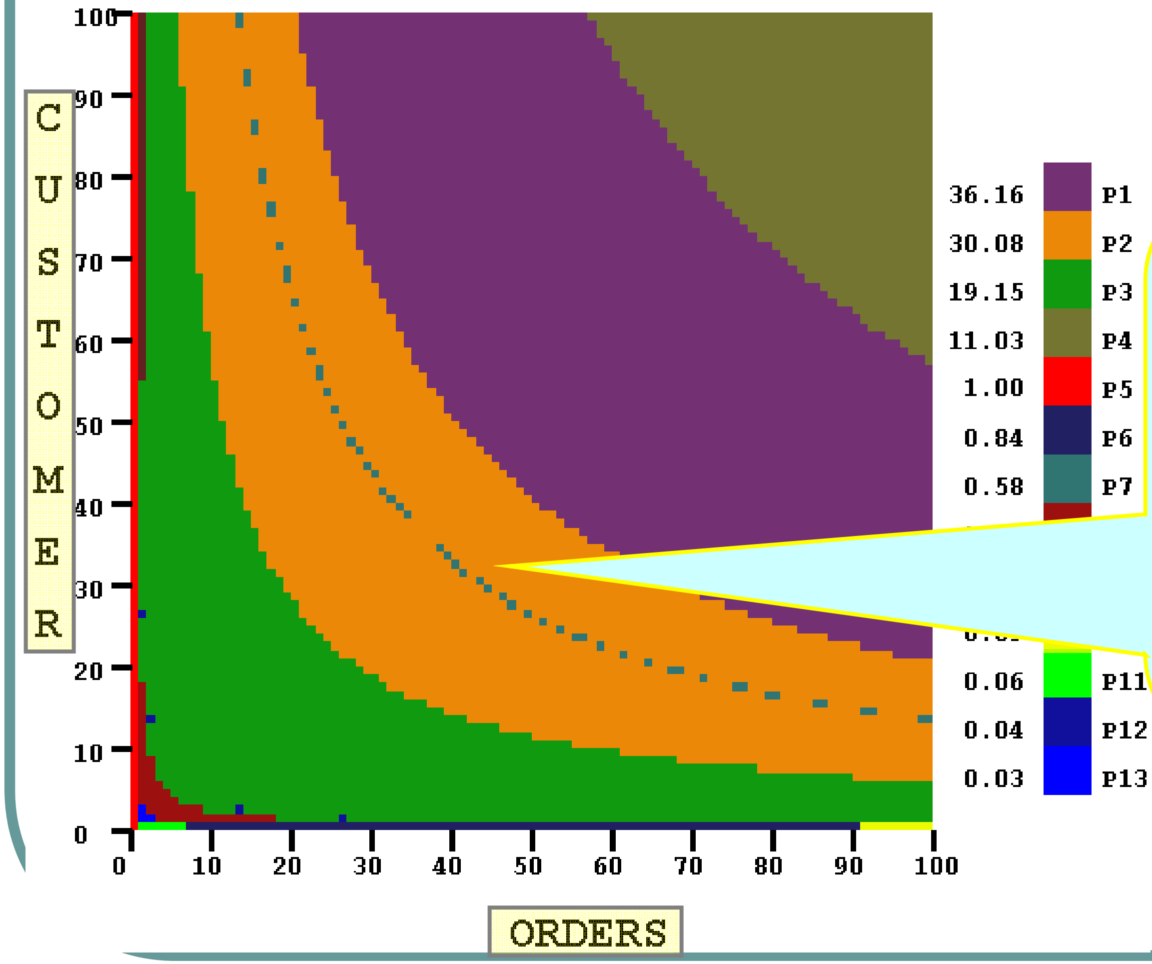


Six plans simultaneously change with rapid alternations to produce a “Venetian blinds” effect.

Left-deep hash join across NATION, SUPPLIER and LINEITEM relations gets replaced by a right-deep hash join.

Footprint Pattern

[QT7,OptA]

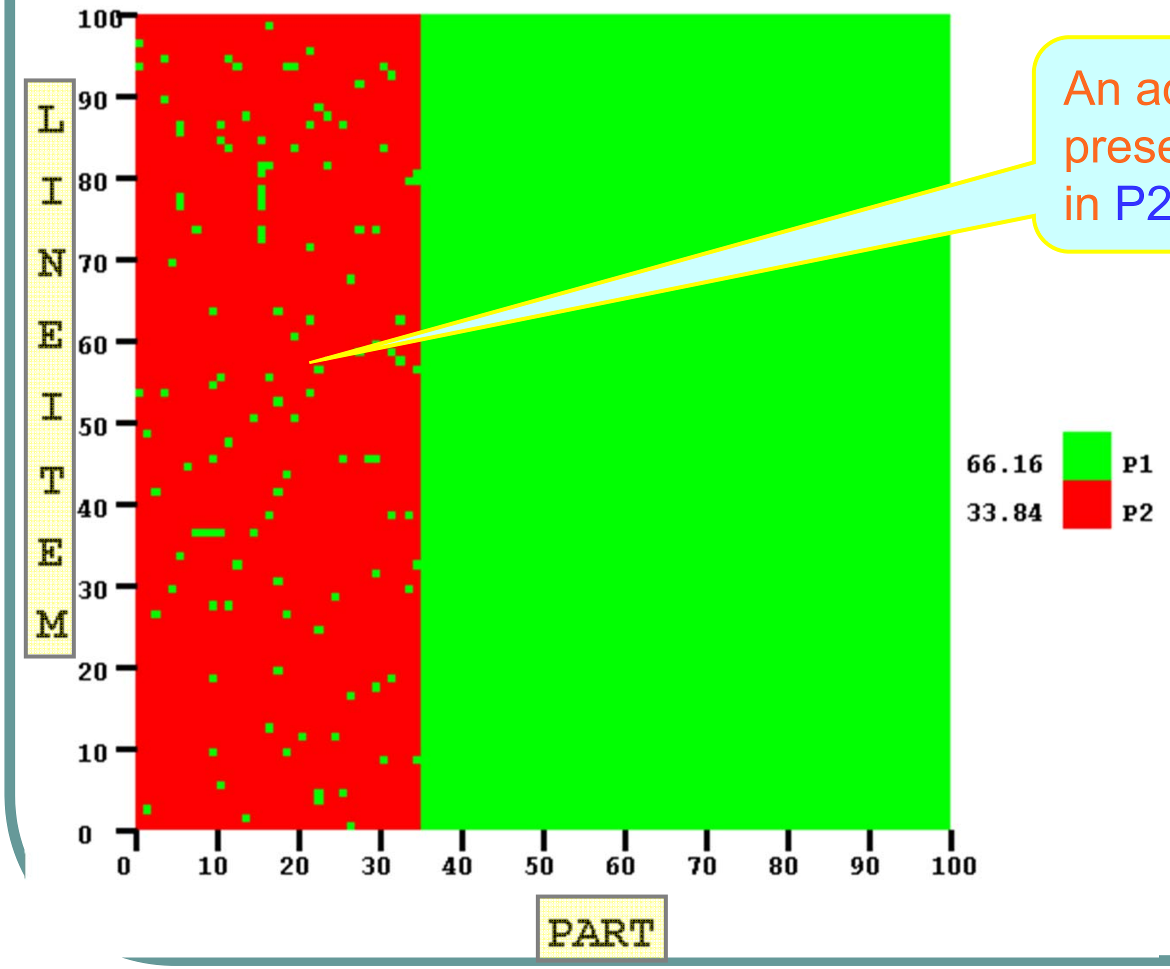


P7 is a thin and broken curved pattern in the middle of P2's region.

P2 has sort-merge-join at the top of the plan tree, while P7 uses hash-join

Speckle Pattern

[QT17,OptA]



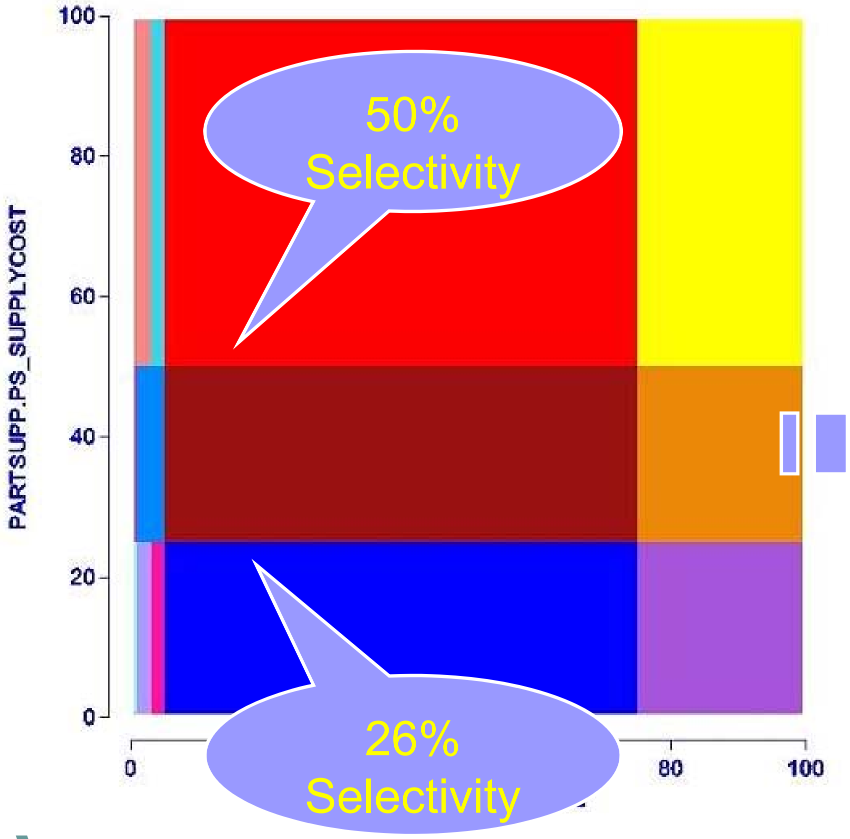
An additional sort operation is present on the PART relation in P2, whose cost is very low

Non-Monotonic Cost Behavior

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

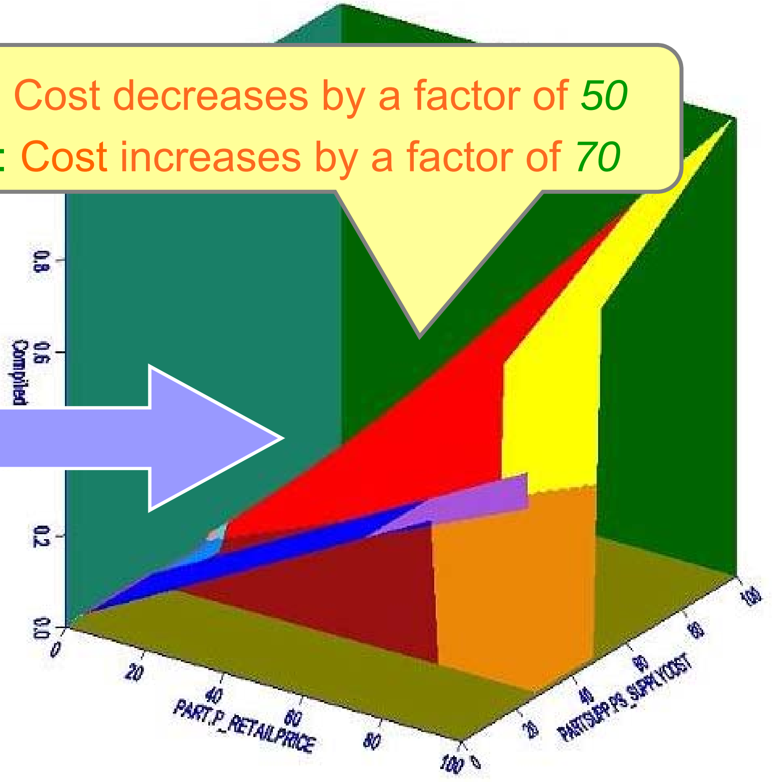
Plan-Switch Non-Monotonic Costs

[QT2,OptA]



50% Selectivity

26% Selectivity



26%: Cost decreases by a factor of *50*

50%: Cost increases by a factor of *70*

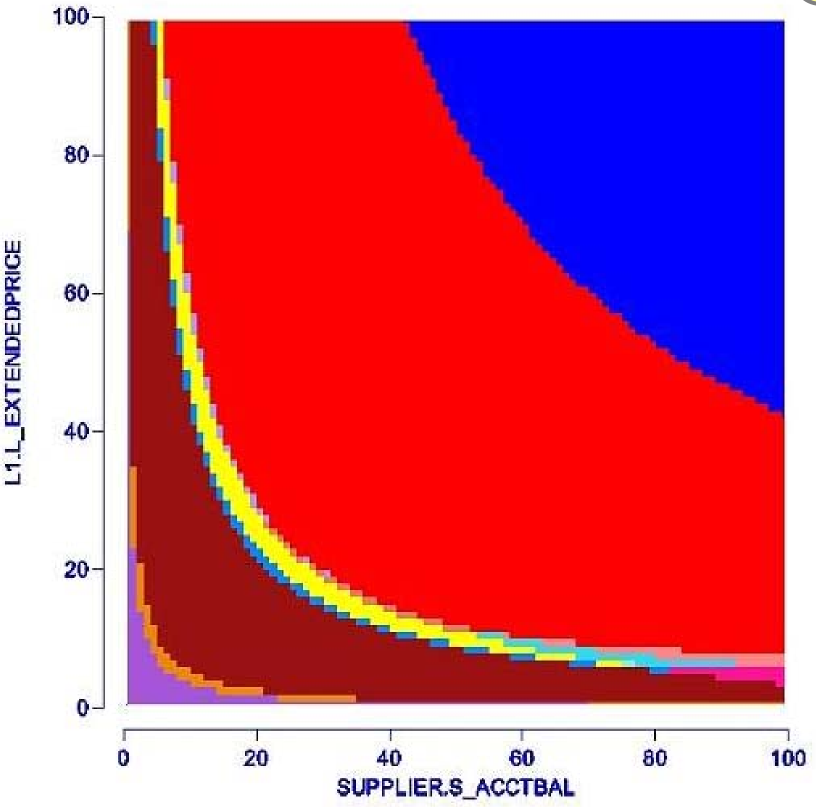
Plan Diagram

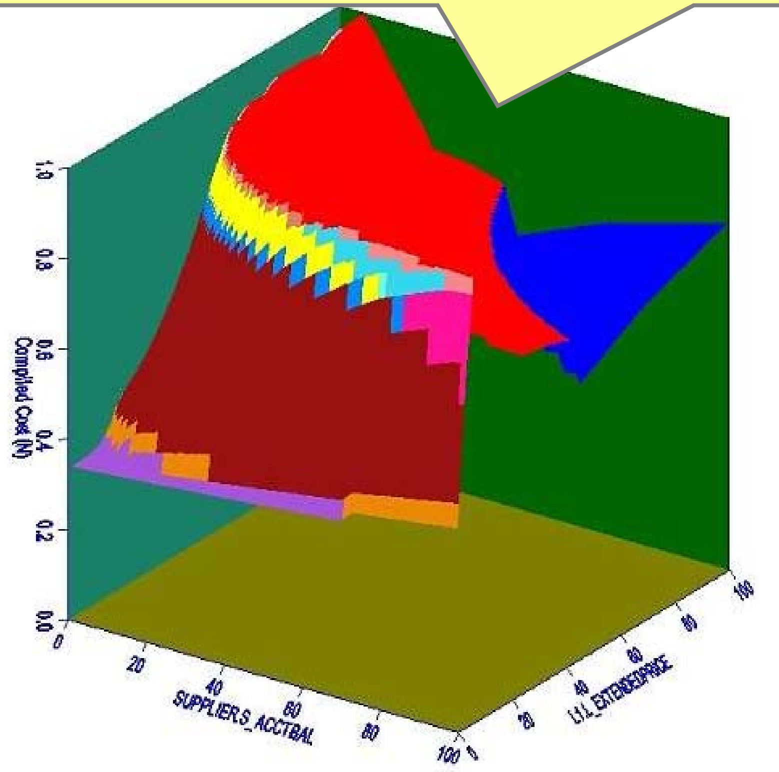
Cost Diagram

Intra-Plan Non-Monotonic Costs

[QT21,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

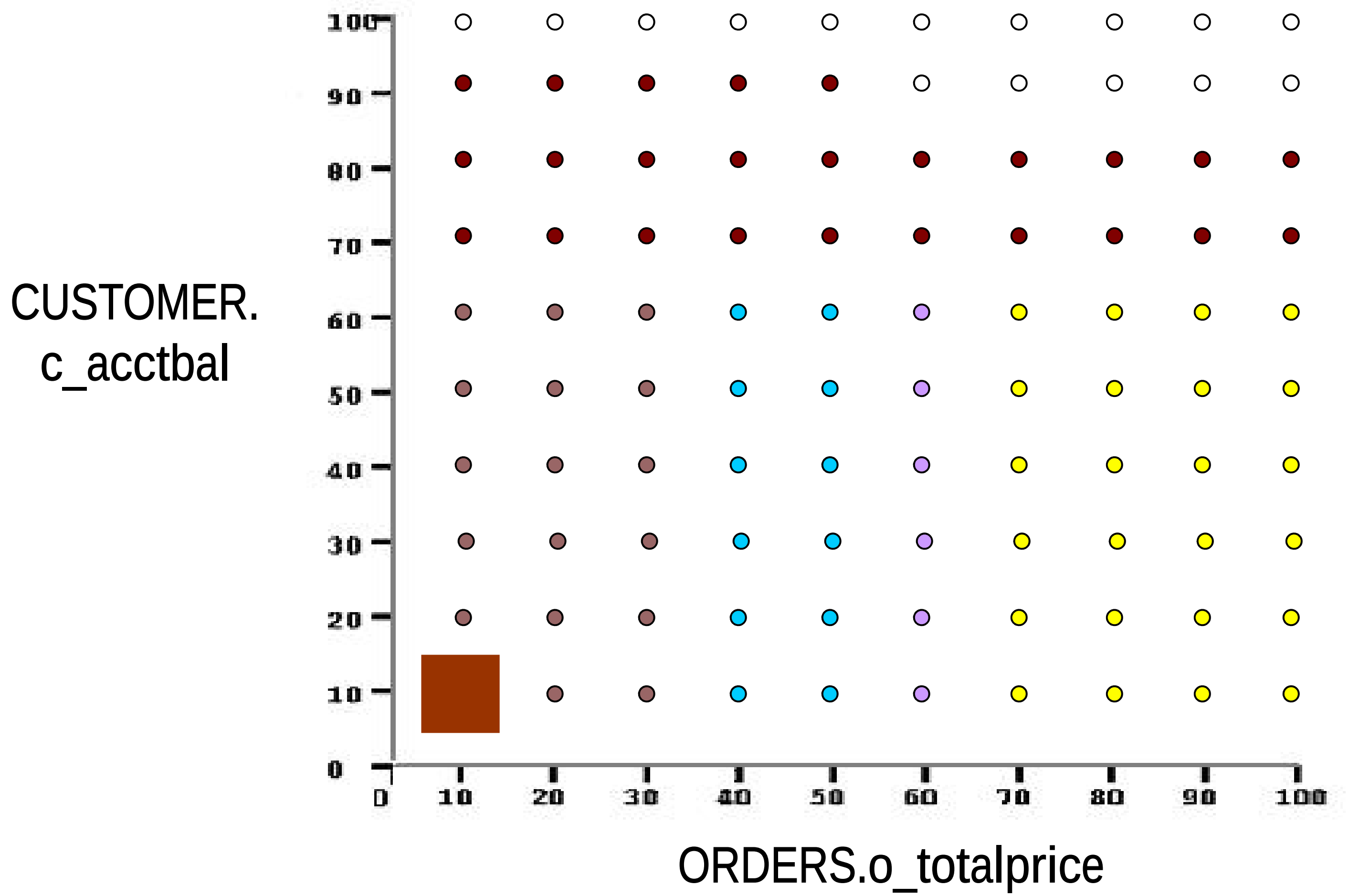


Diagram Generation Overheads

- Generating a 2D plan diagram at resolution 1000, or 3D at resolution 100, requires 10^6 optimizations
- Cost of each optimization: ~ 0.5 sec
- Running time: ~ 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

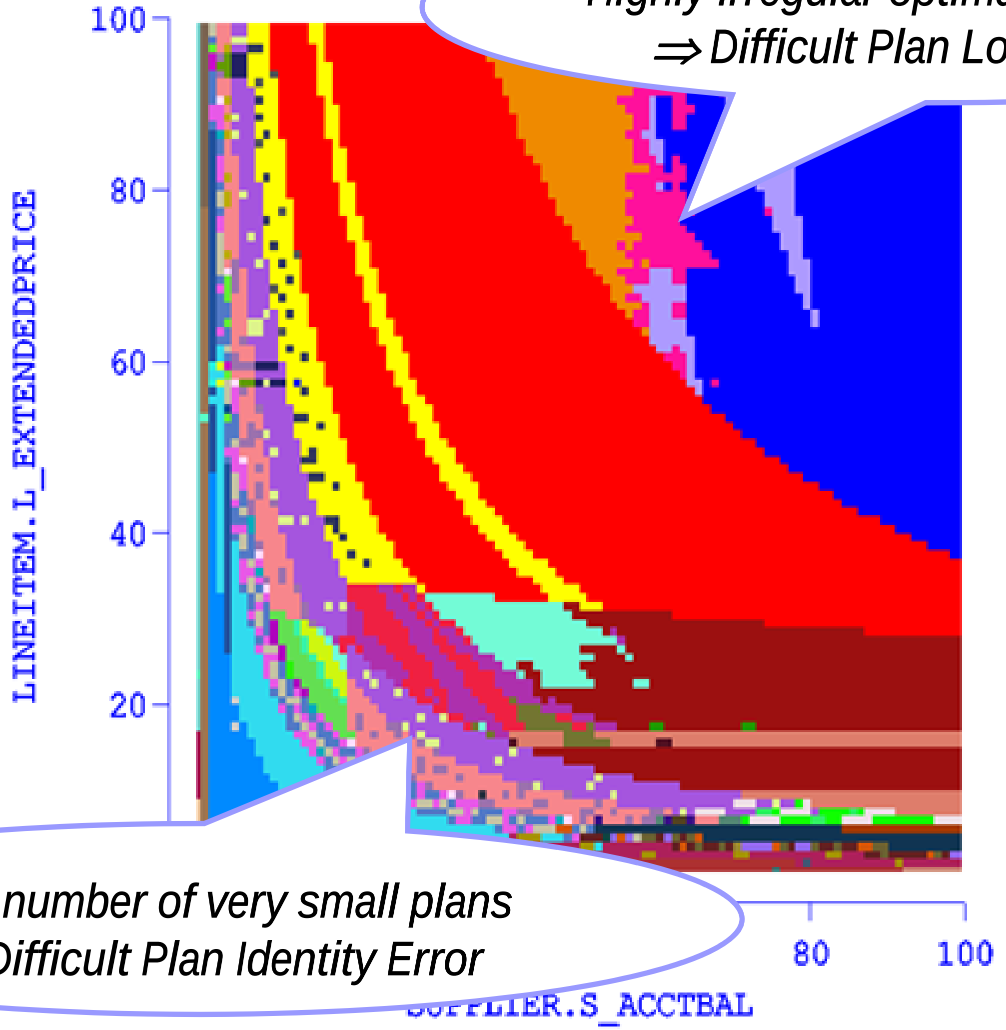
- Plan Identity Error ($\boldsymbol{\varepsilon_I}$) : % of plans that remained unidentified in **A** relative to **P**

$$\epsilon_I = \frac{|P| - |A|}{|P|} \times 100$$

- Plan Location Error ($\boldsymbol{\varepsilon_L}$) : % of points assigned wrong plan in **A** relative to **P**

$$\epsilon_L = \frac{|p_A(q) \neq p_P(q)|}{m} \times 100$$

Road Blocks



Highly irregular optimality boundaries
⇒ Difficult Plan Location Error

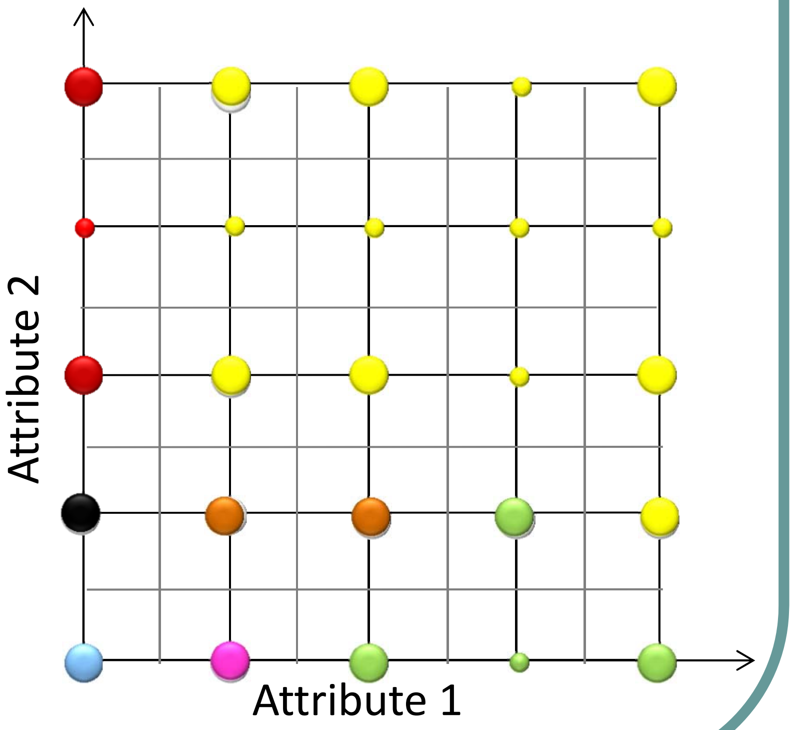
Large number of very small plans
⇒ Difficult Plan Identity Error

SOLUTION TECHNIQUES

- Purely Statistical:
  - Random Sampling with Nearest Neighbor Inferencing
- DB-conscious:
  - Grid Sampling with Parametric Query Optimization (GS_PQO)

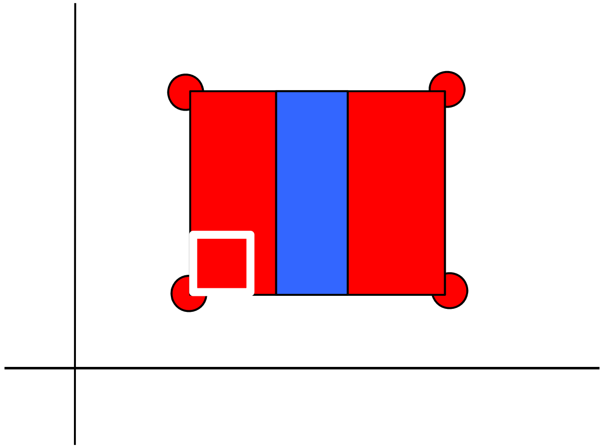
Basic Grid Sampling

- 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

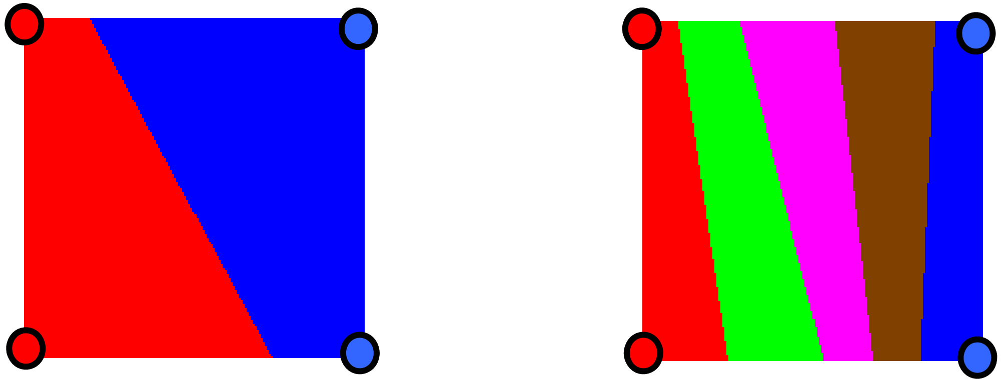
- PQO principle: If two points in a query parameter space have the same optimal plan, then this plan is optimal at all points on the straight line joining them.



- Plan Diagrams severely violate PQO [Part I]
- But, PQO usually holds in **micro-regions**

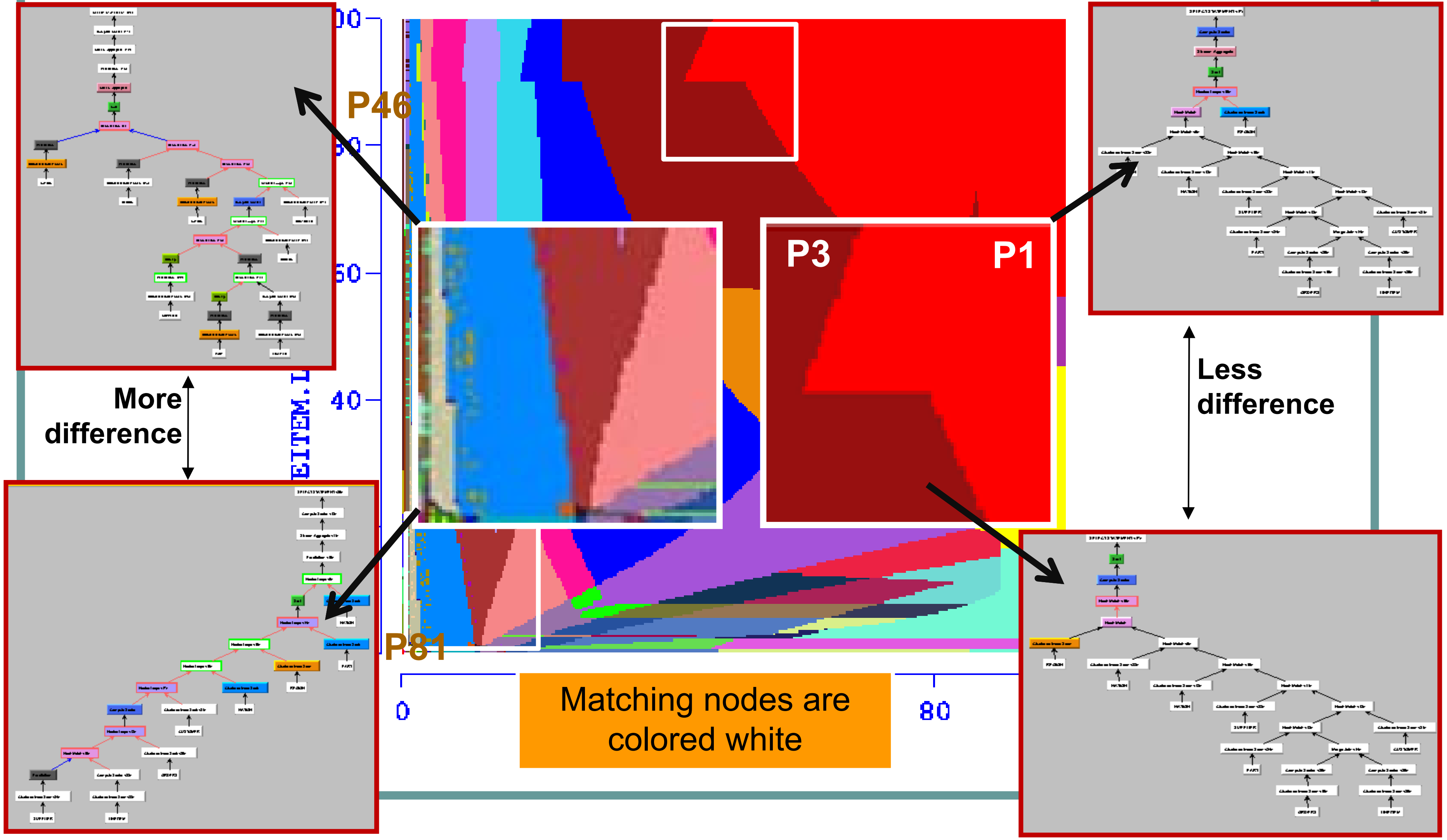
Issues with Basic Grid Sampling

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



  - 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

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

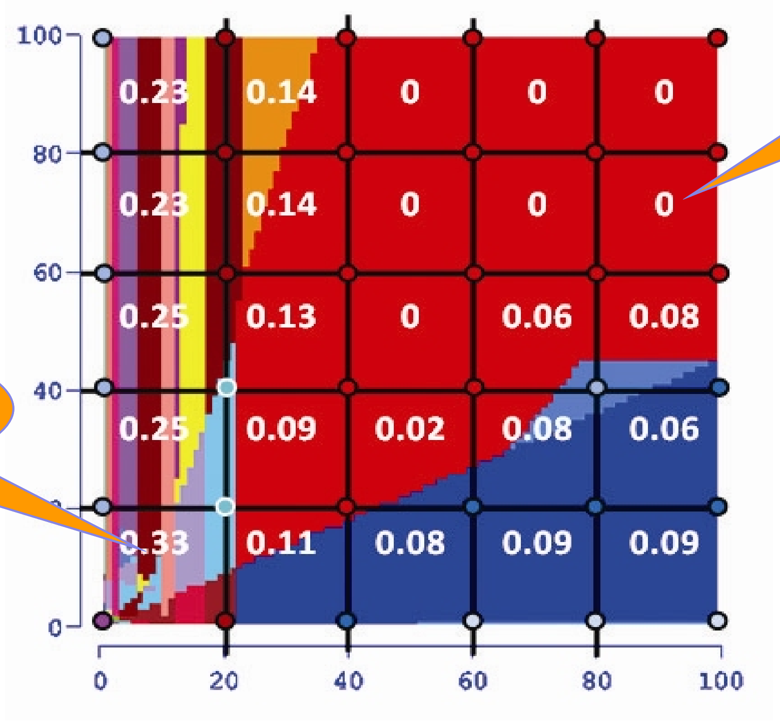
- Hyper-rectangle with n corner points and plan trees $T_1, T_2 \ldots T_n$. Then, overall Plan Density factor is estimated as

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

Plan Density Example

ρ is a metric normalized to [0,1]

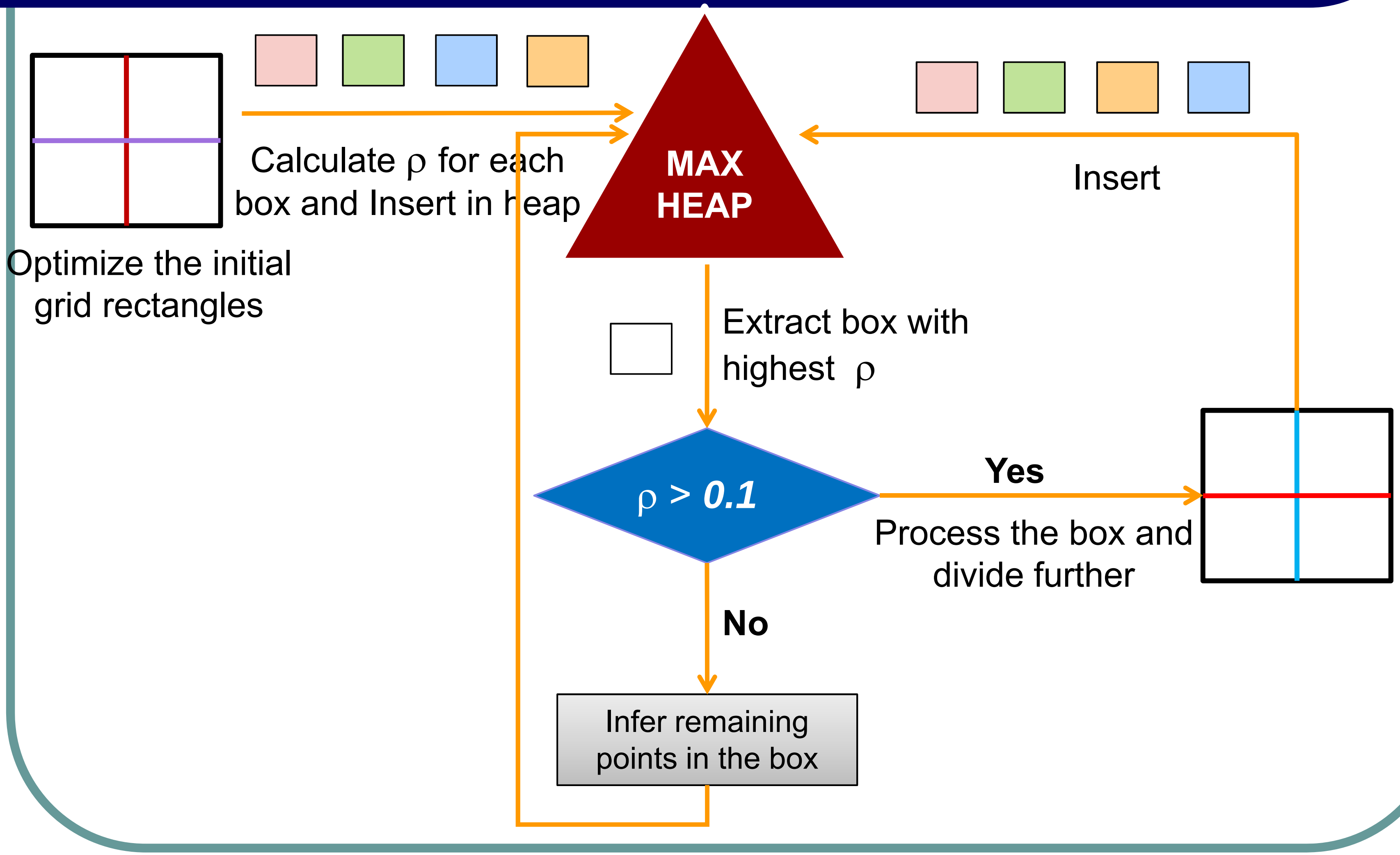
- ρ close to 0 indicates similar plan trees
- ρ close to 1 indicates extremely dissimilar plan trees



Indication of low plan density

Indication of high plan density

Complete GS_PQO



Optimize the initial grid rectangles

Calculate ρ for each box and Insert in heap

MAX HEAP

Insert

Extract box with highest ρ

$\rho > 0.1$

Yes

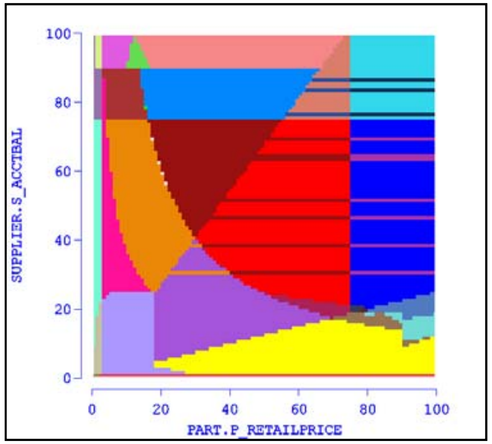
Process the box and divide further

No

Infer remaining points in the box

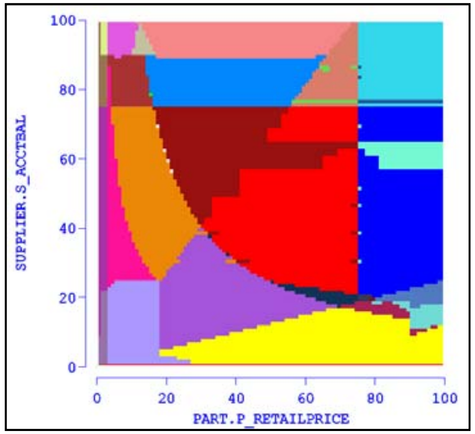
Complex Diagram Approximation Examples

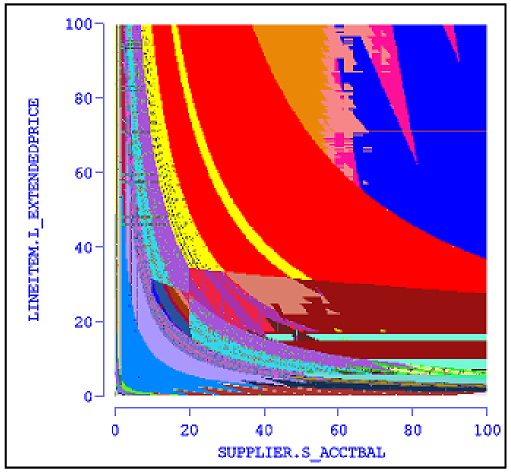
GS_PQO



Sample size 7%

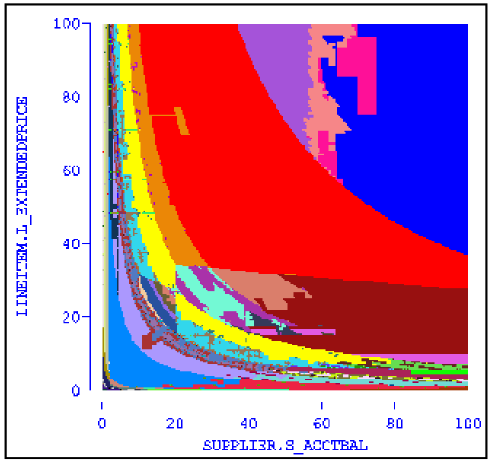
$\varepsilon_I, \varepsilon_L < 10\%$





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

Problem Statement

Can the plan diagram be recolored with a smaller set of colors (i.e. some plans are “swallowed” by others), such that

Guarantee:

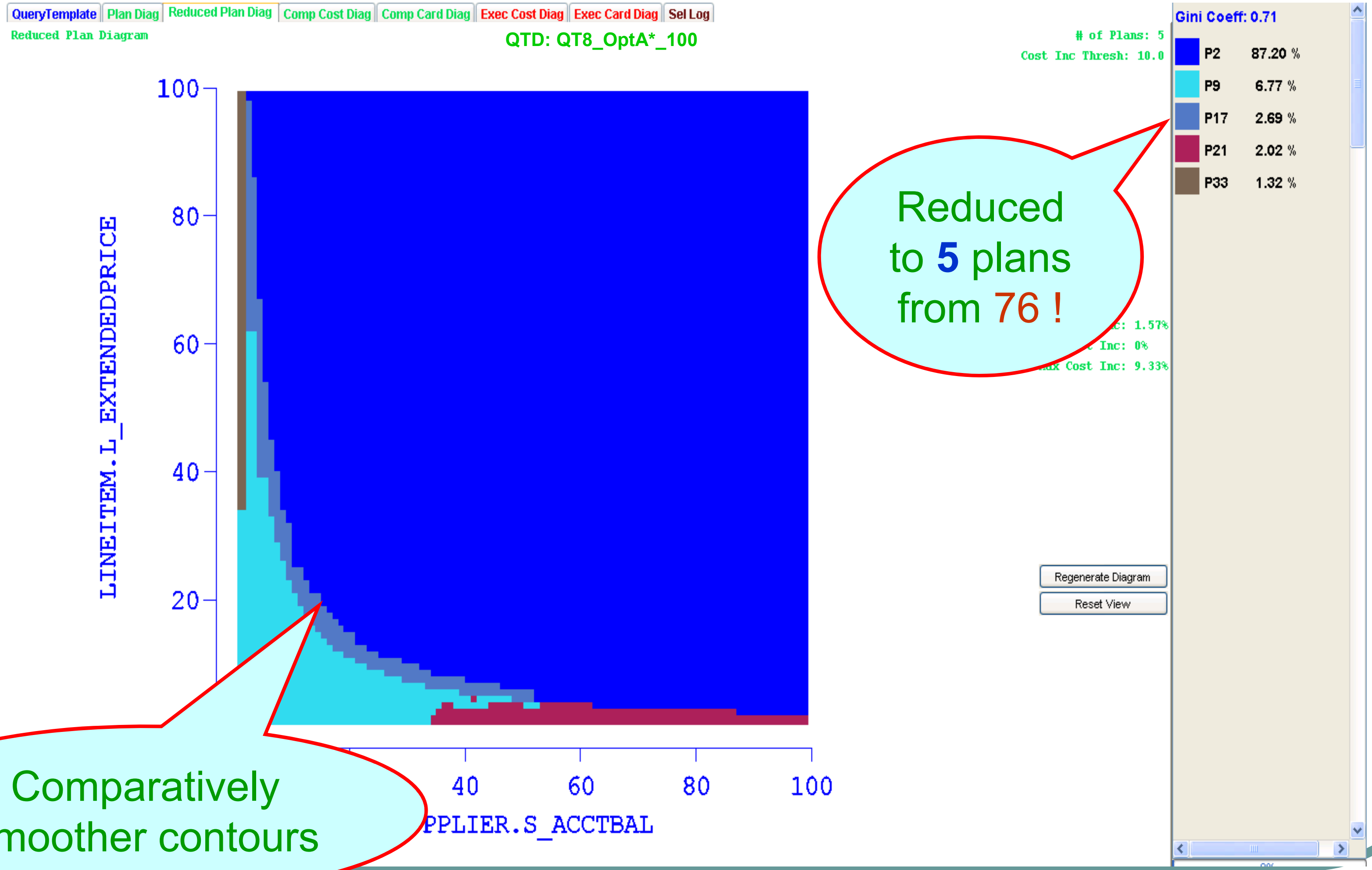
No query point in the original diagram has its estimated cost increased, post-swallowing, by more than λ percent (user-defined)

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 λ percent

Reduced Plan Diagram [λ=10%]

[QT8, OptA*, Res=100]



Reduced Plan Diagram

QTD: QT8_OptA*_100

# of Plans: 5
Cost Inc Thresh: 10.0

c: 1.57%
Inc: 0%
ax Cost Inc: 9.33%

Gini Coeff: 0.71

🟦	P2	87.20 %
🟦	P9	6.77 %
🟦	P17	2.69 %
🟥	P21	2.02 %
🟫	P33	1.32 %

Reduced to **5** plans from 76 !

Comparatively smoother contours

PROBLEM ANALYSIS

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 λ% (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, either

(a) $P_k \in$ R *or*

(b) $\forall q \in P_k$, the assigned replacement plan $P_j \in$ R is

s.t. $$\frac{c(P_j, q)}{c(P_k, q)} \leq 1 + \frac{\lambda}{100}$$

e.g. if λ = 10%, $$\frac{c(P_j, q)}{c(P_k, q)} \leq 1.1$$

Basic Requirement

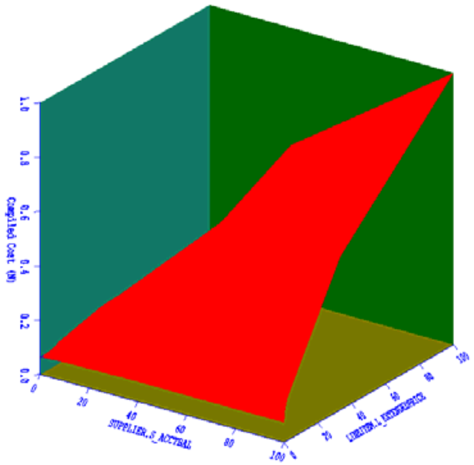
- Need to be able to cost a plan P_k at points outside its own optimality region,
  - called “Foreign Plan Costing” (FPC)

- 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**.

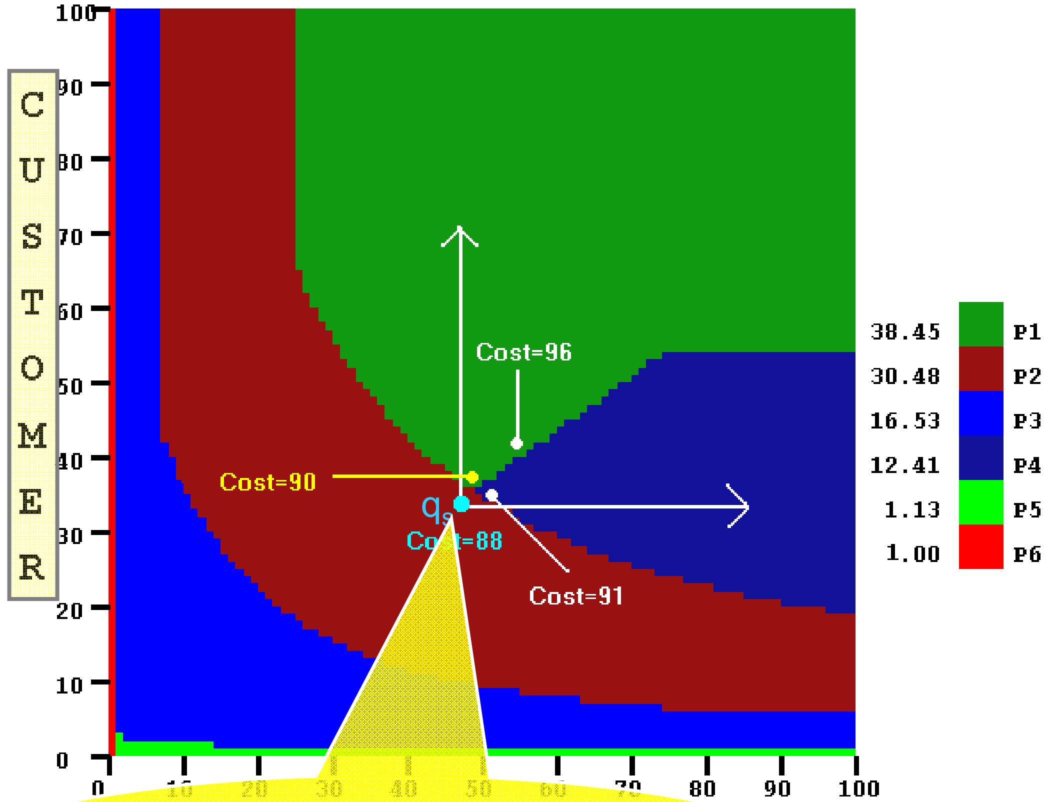


Cost function of plan P_k

True for most query templates since

selectivity↑ ⇒ input data↑ ⇒ query processing↑ ⇒ (est) cost↑

Cost-upper-bounding Approach

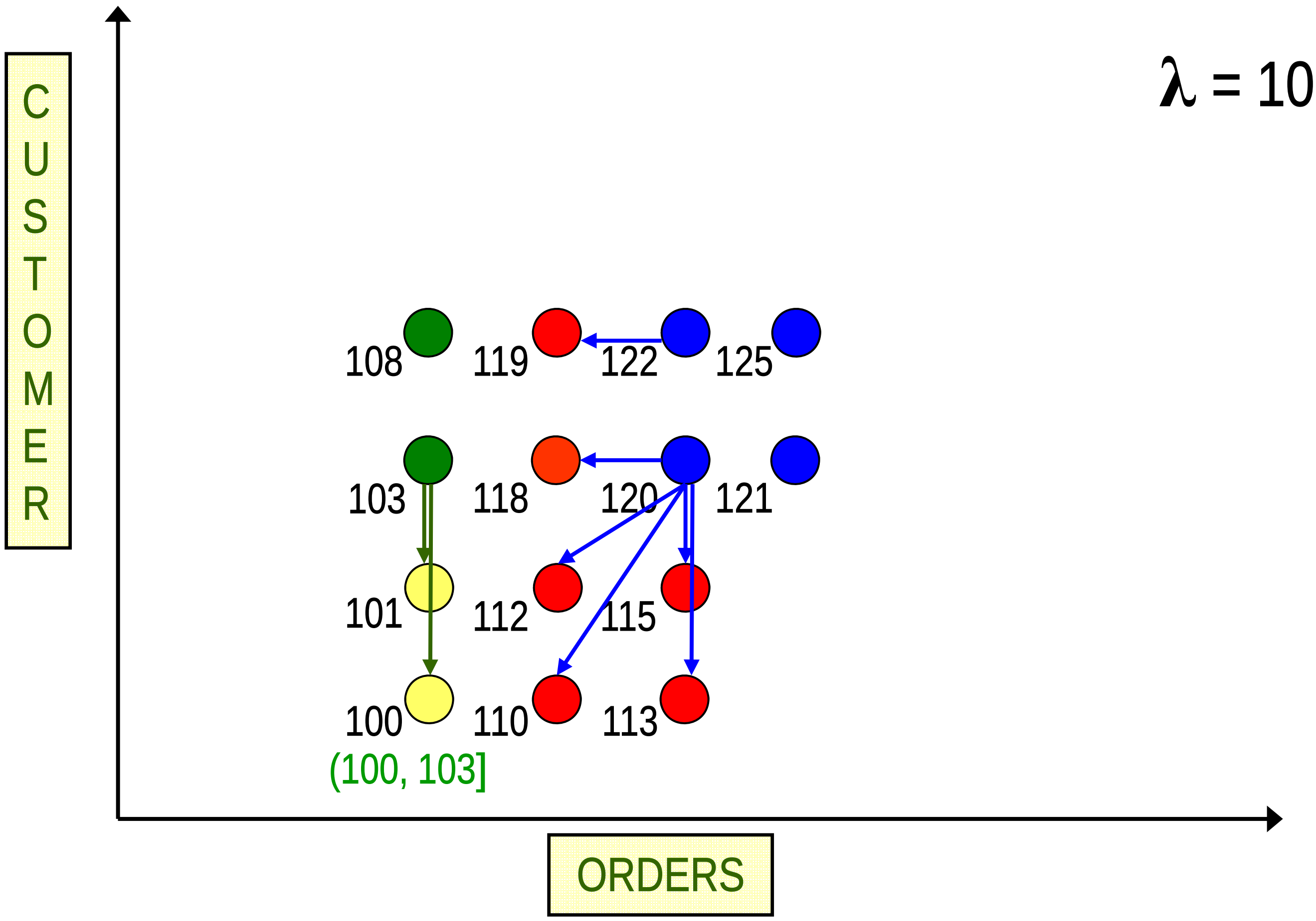


Cost of query point q_s with optimal plan P_2 is 88

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 **P1** or **P4** lies in the intervals **(88, 90]** and **(88,91]**, respectively.

Example Plan Swallowing



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

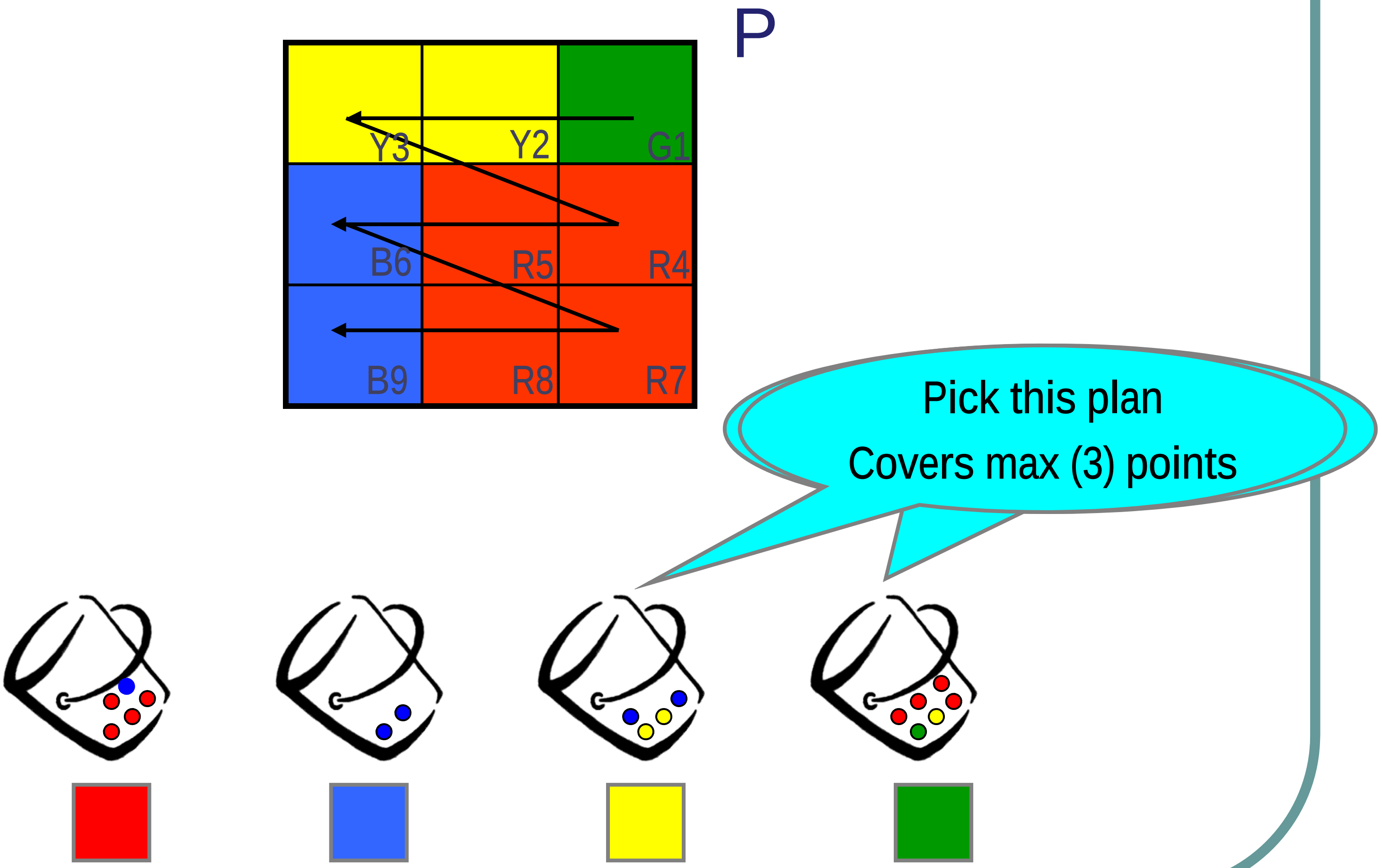
- Assign a bin to each individual plan in **P**
- Start at the top right corner and proceed in reverse row-major order
  - first-quadrant info available when processing a query point
- Put a copy of each query point into all plan-bins (subsets) that it can belong to w.r.t. λ constraint: SetCover problem
- 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

🟩

🟨



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

[OptC, Res = 30E, λ = 20%]

TPC-H Query Template	Original # of Plans	Reduced Plans CostGreedy	Reduced Plans CG-FPC
QT2	60	14	3
QT5	51	7	2
QT8	121	7	2
QT9	137	9	3
QT10	44	3	3

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

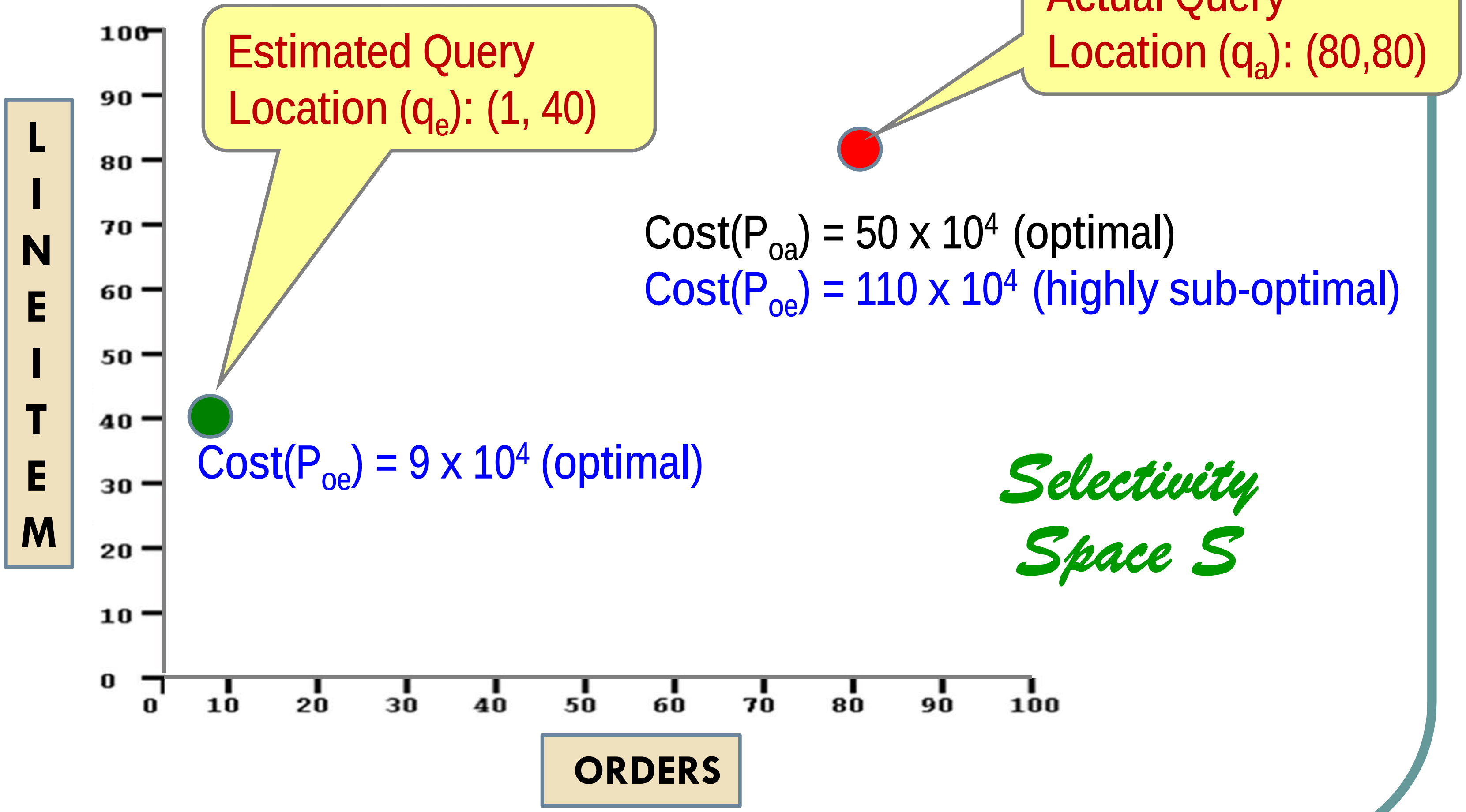
$q_e(x_e, y_e)$: estimated location by optimizer

$q_a(x_a, y_a)$: 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)

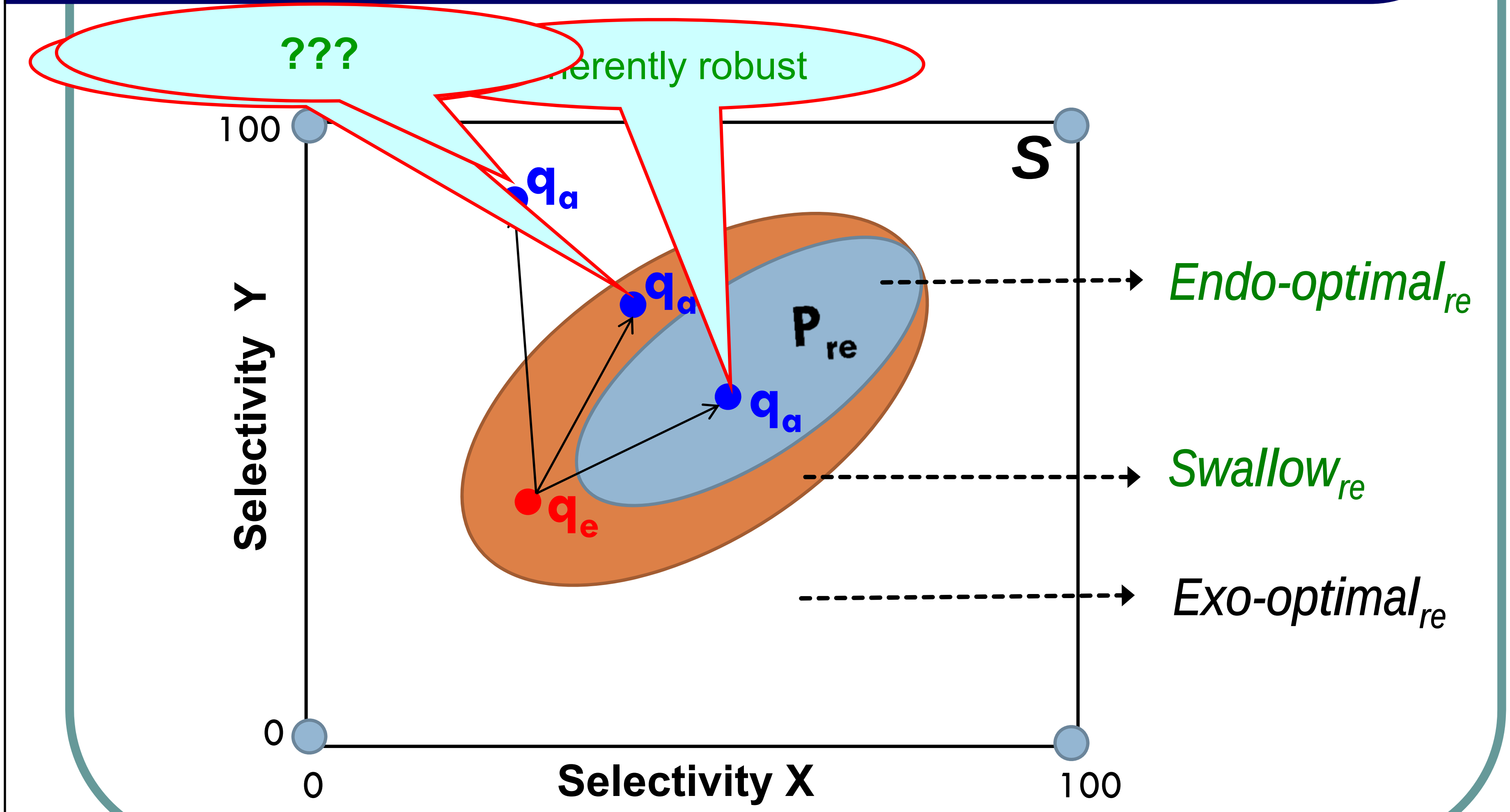
Cost(P_{oa}) = 50 x 10^4 (optimal)

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

Cost(P_{oe}) = 9 x 10^4 (optimal)

Selectivity Space S

Error Locations wrt Plan Replacement Regions

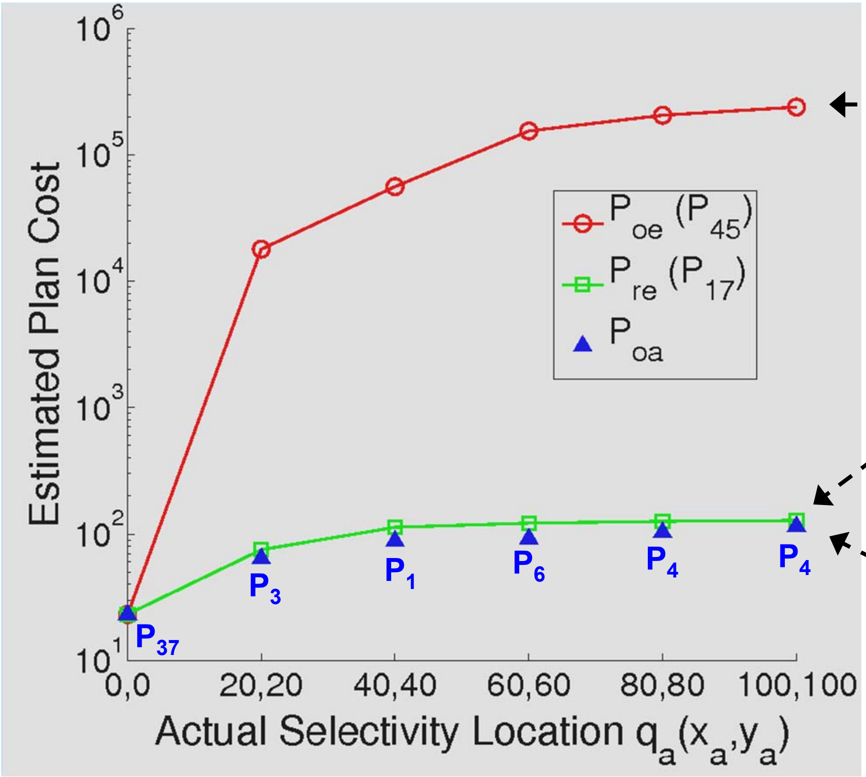


Positive Impact of Reduction

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

QT5

$q_e = (0.36, 0.05)$



Original Plan (P_{oe})

Replacement Plan (P_{re})

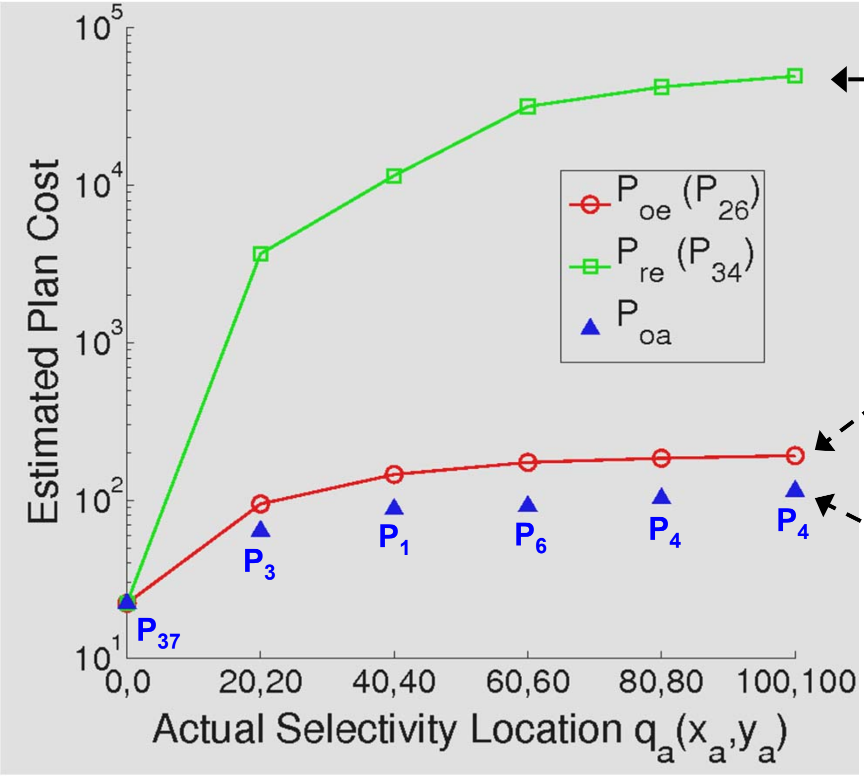
Local Optimal Plan (P_{oa})

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

  - ∀ points q in P_{oe}'s endo-optimality region,

    $c(P_{re}, q) \leq (1+\lambda)\, c(P_{oe}, q)$

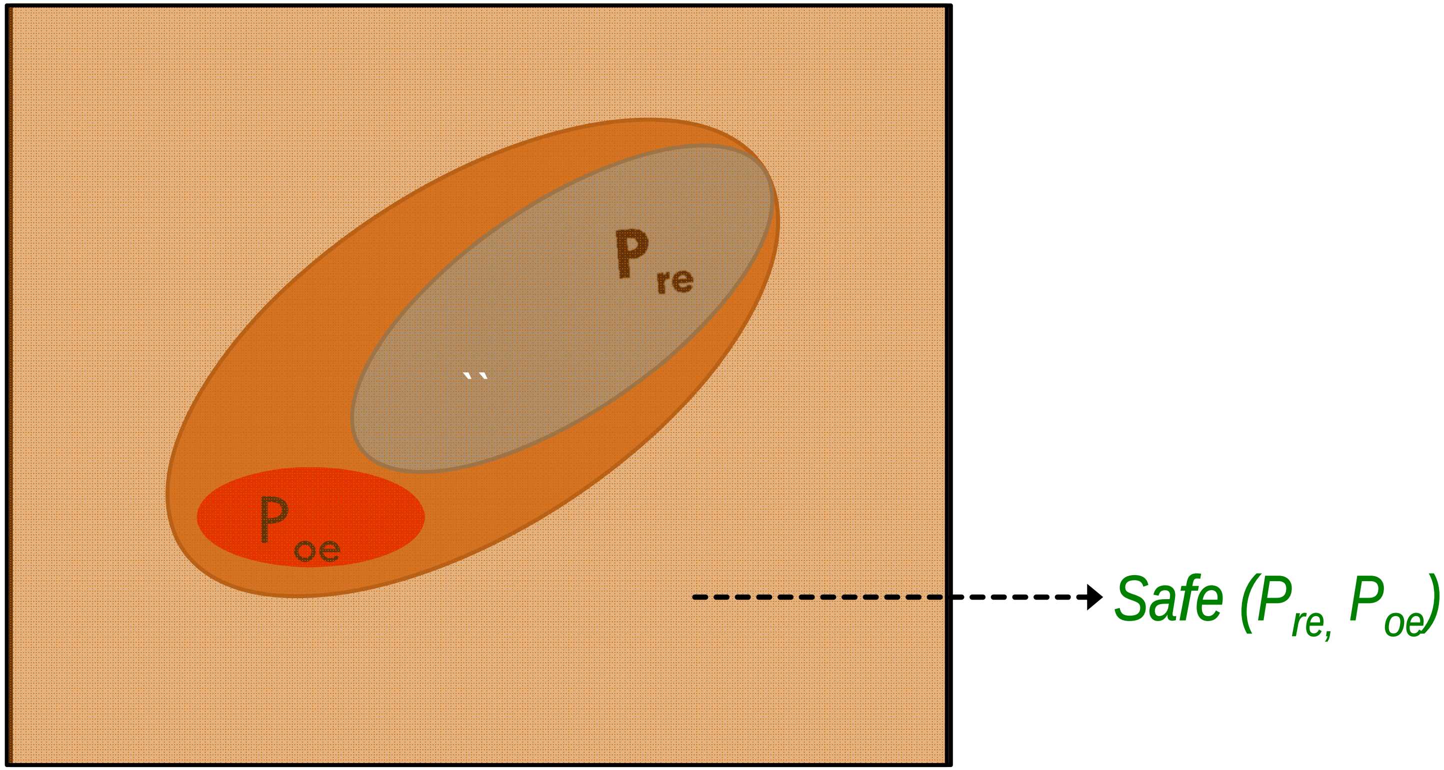
- New global constraint:

  P_{re} can replace P_{oe} only if it guarantees a globally safe space

  - ∀ points q in selectivity space S,

    $c(P_{re}, q) \leq (1+\lambda)\, c(P_{oe}, q)$

Globally Safe Replacement



Safe ($P_{re,}$ P_{oe})

Analogy Update

USA can annex Cuba only if American passport can guarantee cost of living of Cuban citizen is within λ 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 s c
current optimizers, we can it:

$$PlanCost_P(x,y) = a_1 x + a_2 y + a_3 xy + a_4 x \log x + a_5 y \log y + a_6 xy \log xy + a_7$$

Index Scan
Aggregate

Join

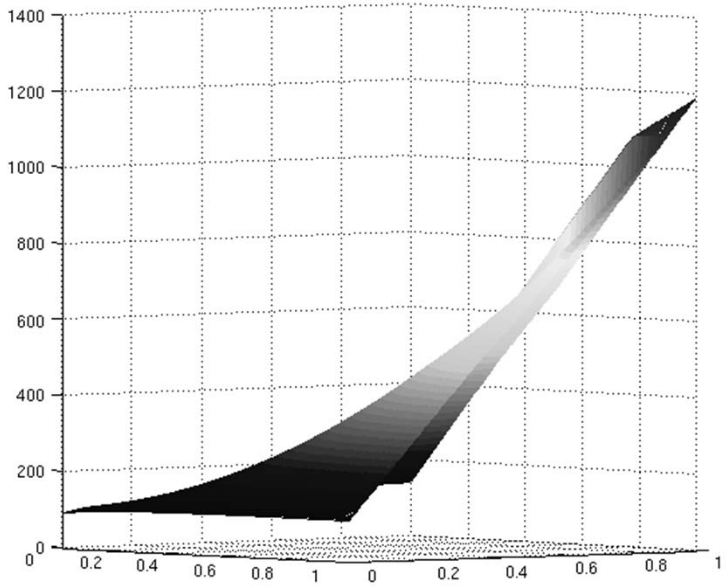
Sort
Group

TableScan

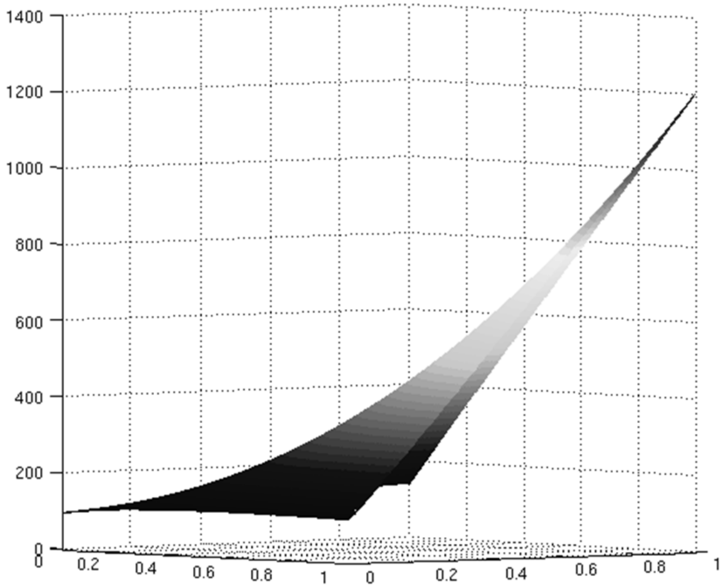
The specific values of a_1 through a_7 are a function of P .

Extension to n-dimensions is straightforward

Cost Model Fit Example



Original Cost Function



Fitted Cost Function

$$Cost(x,y) = 17.9x + 45.9y + 1046xy - 39.5x\log x + 4.5y\log y + 27.6xy\log xy + 97.3$$

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 ⇒ Interior Safety !

Safe and Violating Points

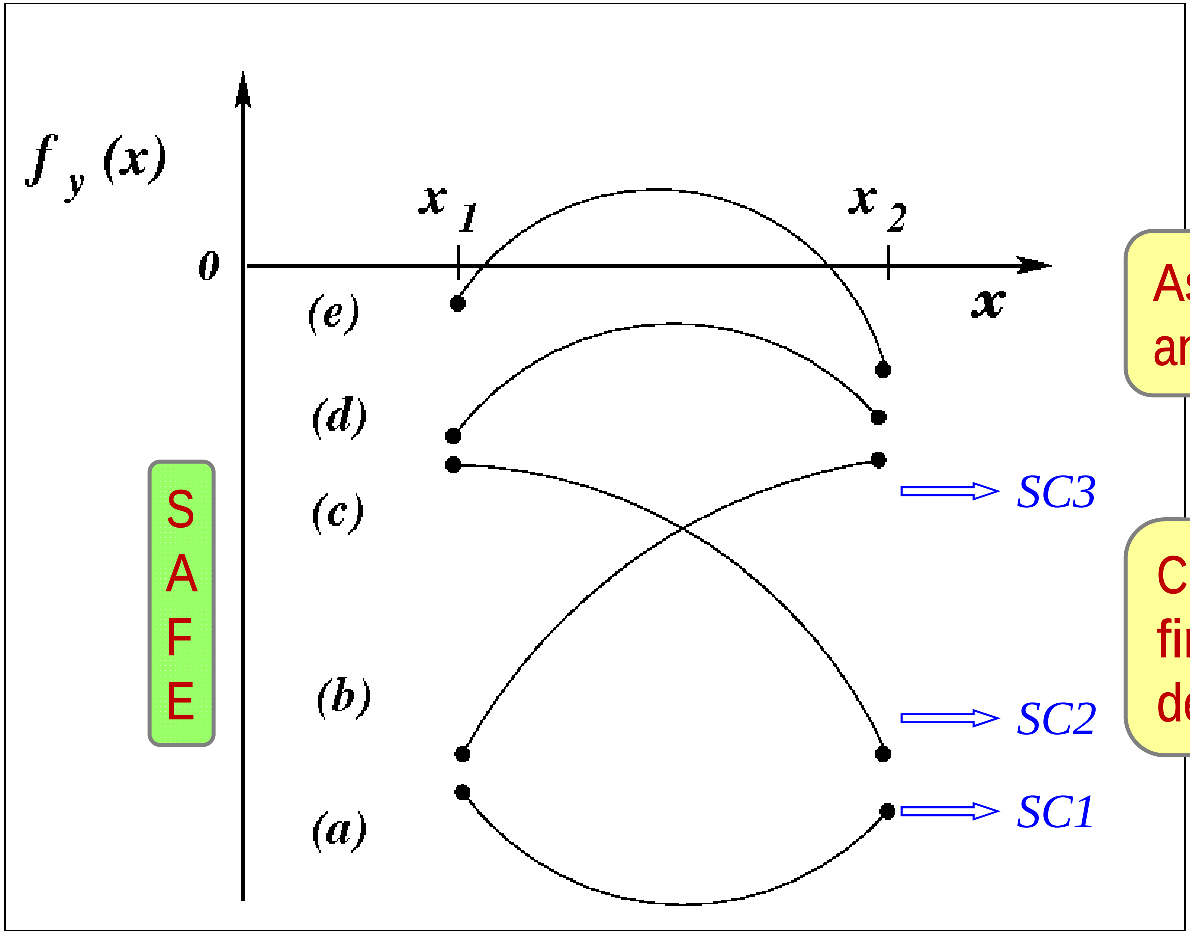
- $f_{oe}(x,y)$: cost function of P_{oe}
- $f_{re}(x,y)$: cost function of P_{re}
- Safety Function

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

- 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

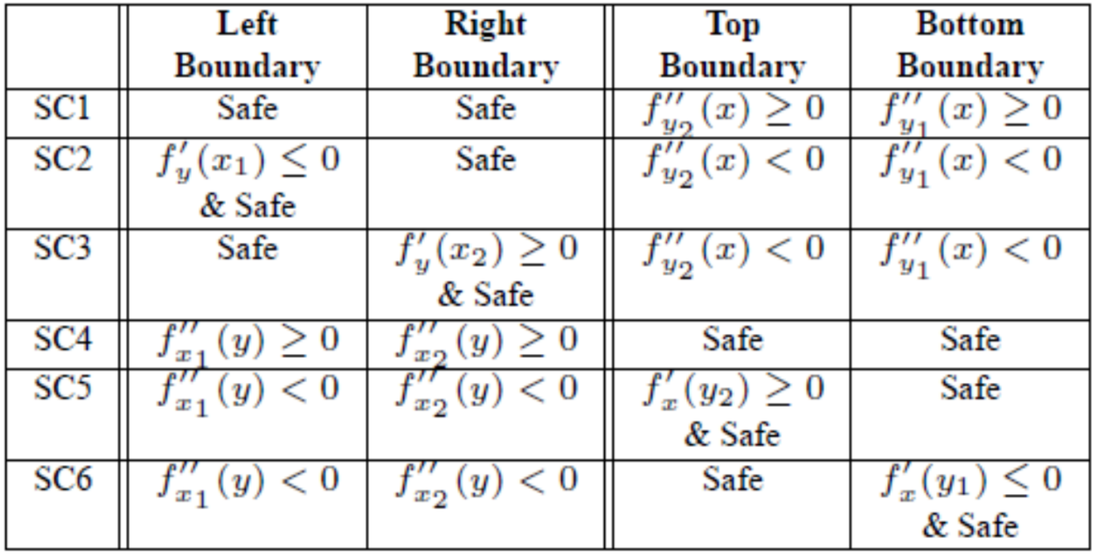


Assume both (d) and (e) are unsafe

Checked through first and second derivatives

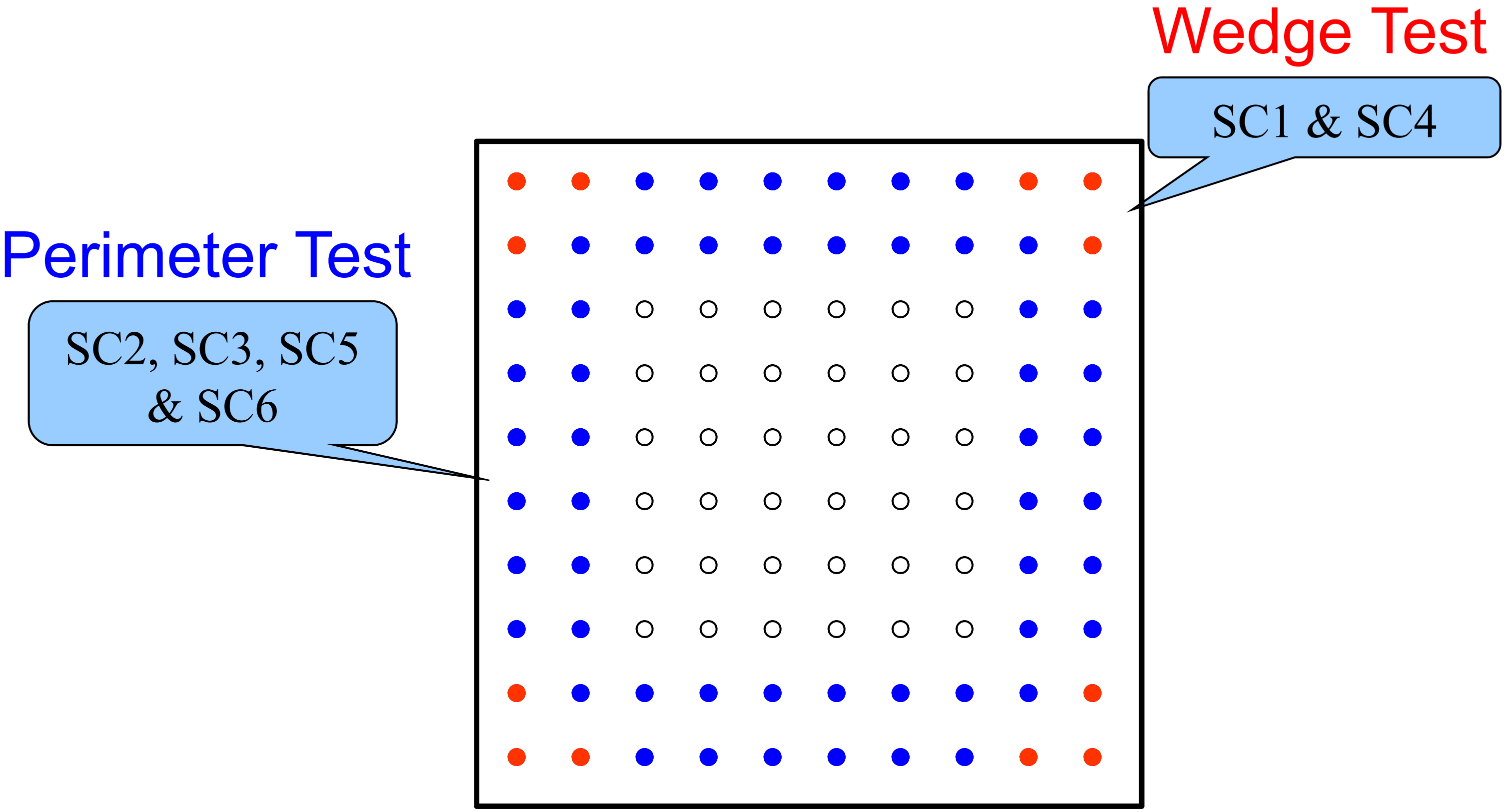
Safety Check Theorem

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



	Left Boundary	Right Boundary	Top Boundary	Bottom Boundary
SC1	Safe	Safe	$f''_{y_2}(x) \geq 0$	$f''_{y_1}(x) \geq 0$
SC2	$f'_y(x_1) \leq 0$ & Safe	Safe	$f''_{y_2}(x) < 0$	$f''_{y_1}(x) < 0$
SC3	Safe	$f'_y(x_2) \geq 0$ & Safe	$f''_{y_2}(x) < 0$	$f''_{y_1}(x) < 0$
SC4	$f''_{x_1}(y) \geq 0$	$f''_{x_2}(y) \geq 0$	Safe	Safe
SC5	$f''_{x_1}(y) < 0$	$f''_{x_2}(y) < 0$	$f'_x(y_2) \geq 0$ & Safe	Safe
SC6	$f''_{x_1}(y) < 0$	$f''_{x_2}(y) < 0$	Safe	$f'_x(y_1) \leq 0$ & Safe

SafetyCheck Algorithm



Wedge Test

SC1 & SC4

Perimeter Test

SC2, SC3, SC5
& SC6

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

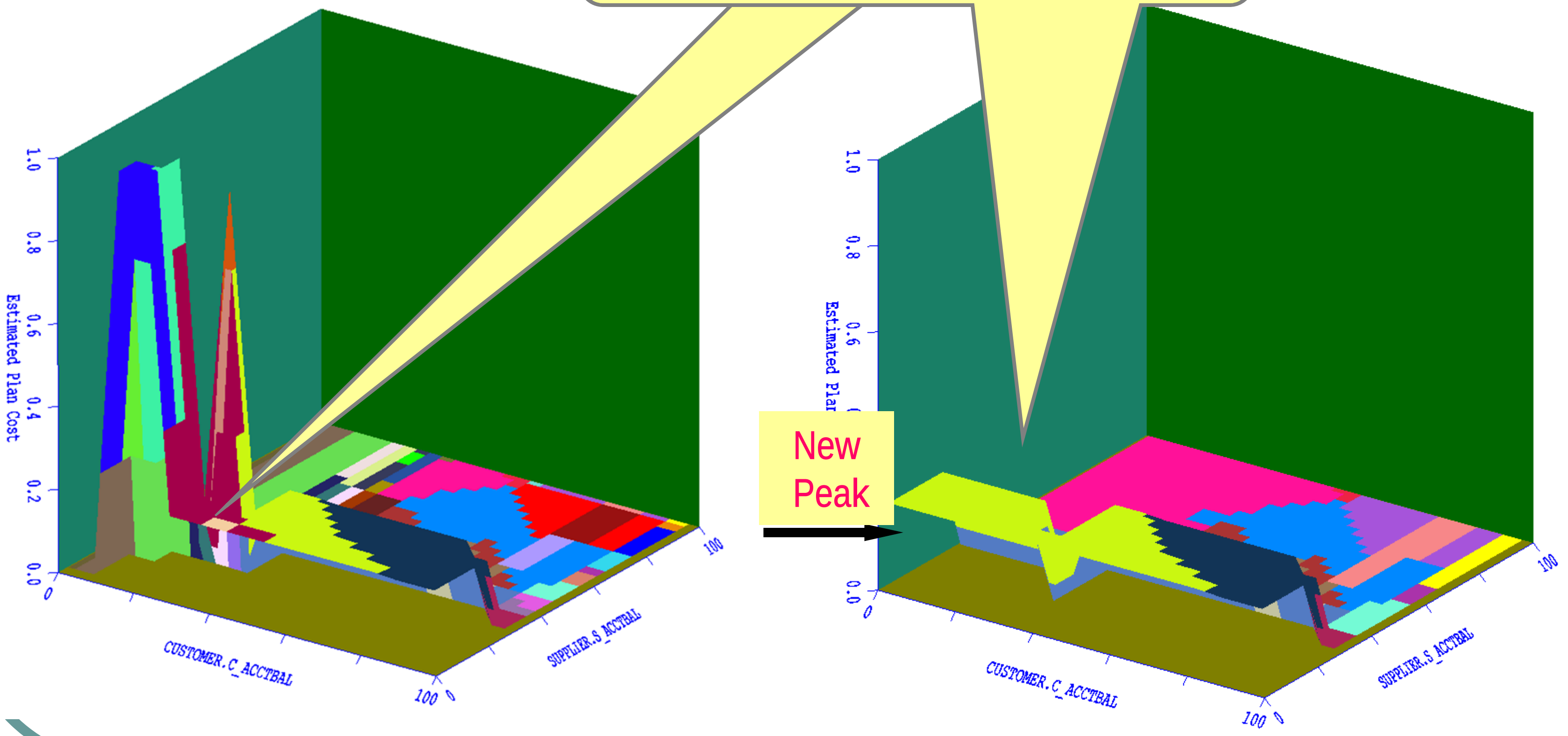
Error Resistance Example

Original (Aggregate)
Cost Diagram

lReduction
duction

Error sensitive locations

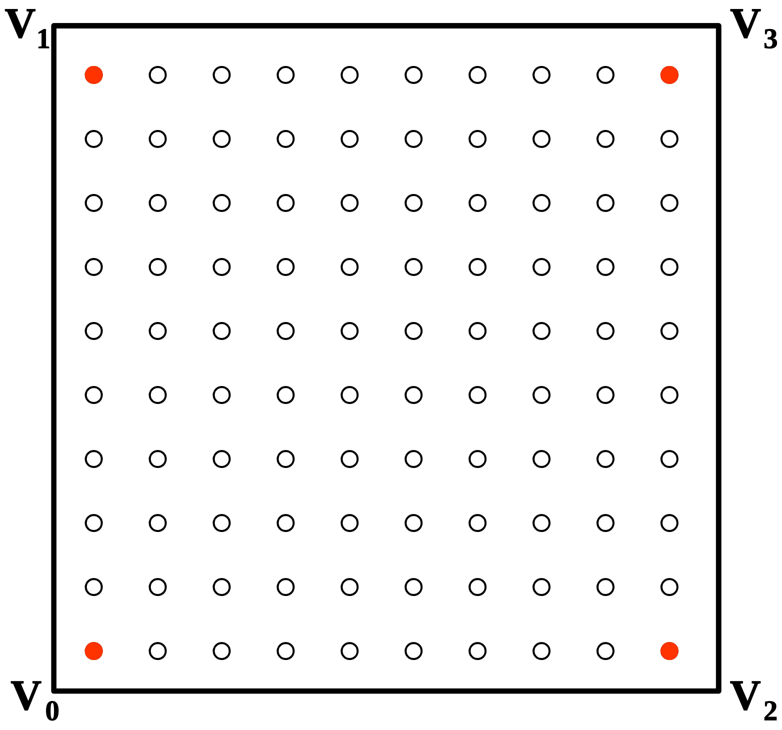
Provides robustness without
introducing any material harm



New
Peak

LiteSEER Heuristic Algorithm

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



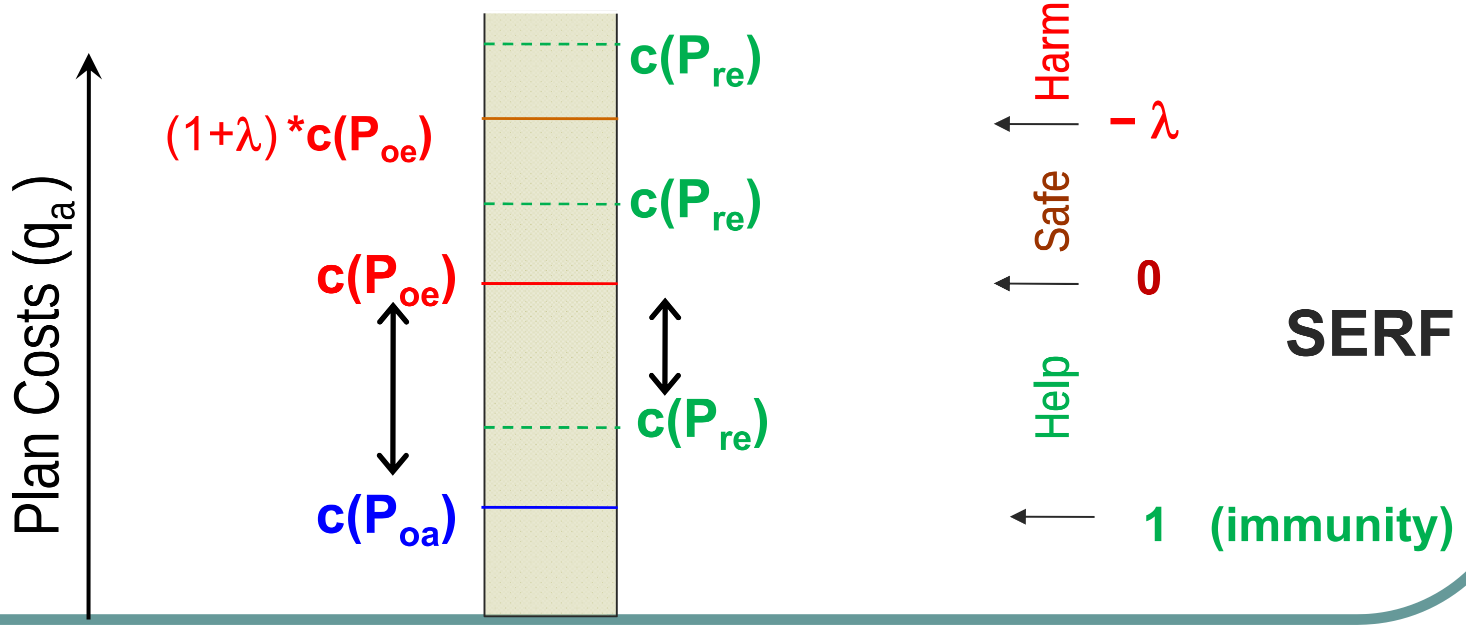
- Time Complexity
  - $O(n^2)$
  - Lower Bound

Measuring Robustness

- Selectivity Error Resistance Factor (SERF)

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

- At location q_a, fraction of performance gap closed by P_{re}



Plan Costs (q_a)

$c(P_{re})$

$(1+\lambda)*c(P_{oe})$

$c(P_{re})$

$c(P_{oe})$

$c(P_{re})$

$c(P_{oa})$

Harm

$-\lambda$

Safe

0

SERF

Help

1 (immunity)

Aggregate Impact of Replacements

$$AggSERF = \frac{\sum_{q_e \in rep(\mathsf{S})} \sum_{q_a \in exo_{oe}(\mathsf{S})} SERF(q_e, q_a)}{\sum_{q_e \in \mathsf{S}} \sum_{q_a \in exo_{oe}(\mathsf{S})} 1}$$

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 atleast 2/3

Robustness Results

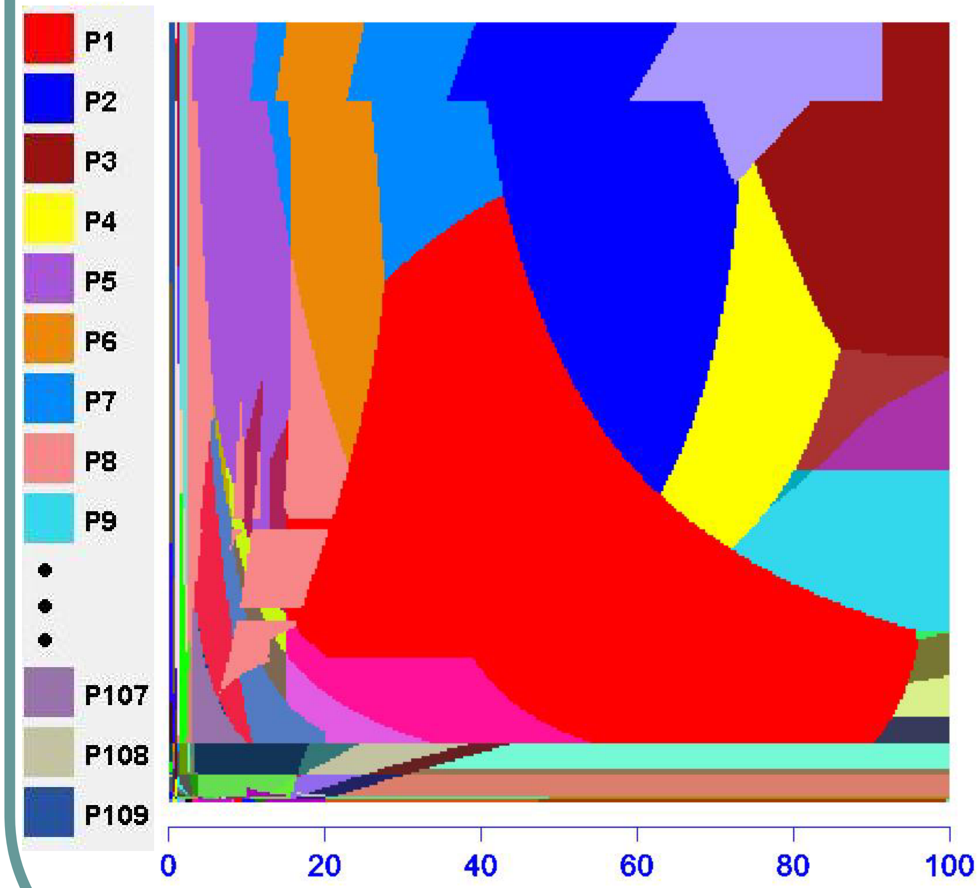
Unsafe replacements

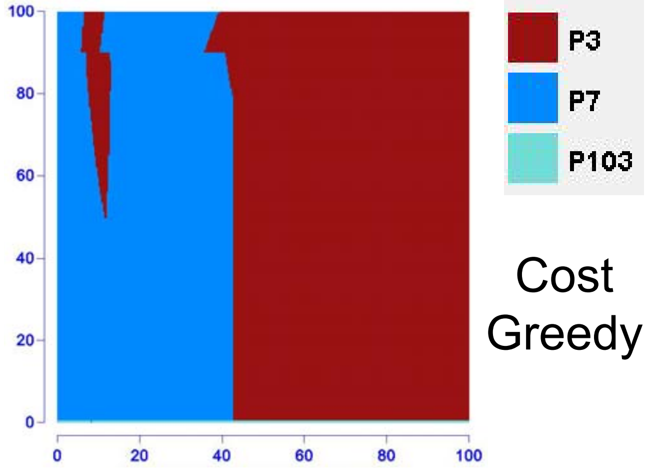
Good robustness + safety + help + anorexia

TPCH QUERY TEMPLATE (TOTAL PLANS)	COST-GREEDY					SEER					LITE-SEER				
	PLANS	MIN SERF	AGG SERF	REP %	HELP %	PLANS	MIN SERF	AGG SERF	REP %	HELP %	PLANS	MIN SERF	AGG SERF	REP %	HELP %
QT2 (60)	14	-60.2	-0.04	54	4	7	-0.8	0.18	90	4	7	-0.8	0.18	90	4
QT5 (51)	7	-15.7	0.24	84	32	2	-0.3	0.29	90	24	2	-0.3	0.29	90	24
QT8 (121)	7	-4.5	0.72	89	78	2	-0.3	0.91	99	92	2	-0.3	0.91	100	92
QT9 (137)	9	-33.6	-0.04	86	22	5	-1.4	0.56	99	48	5	-2.2	0.56	[illegible]	49
QT10 (44)	3	-24.8	-0.24	85	1	3	-1.0	0.13	98	5	3	[illegible]	[illegible]	[illegible]	[illegible]

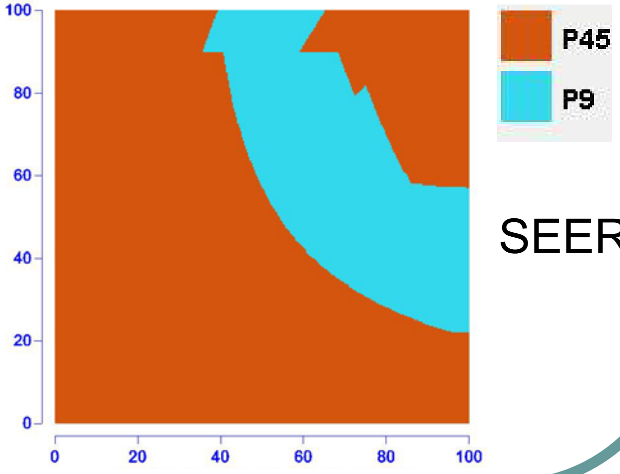
Comparable to SEER

Comparative Reductions





Cost Greedy



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

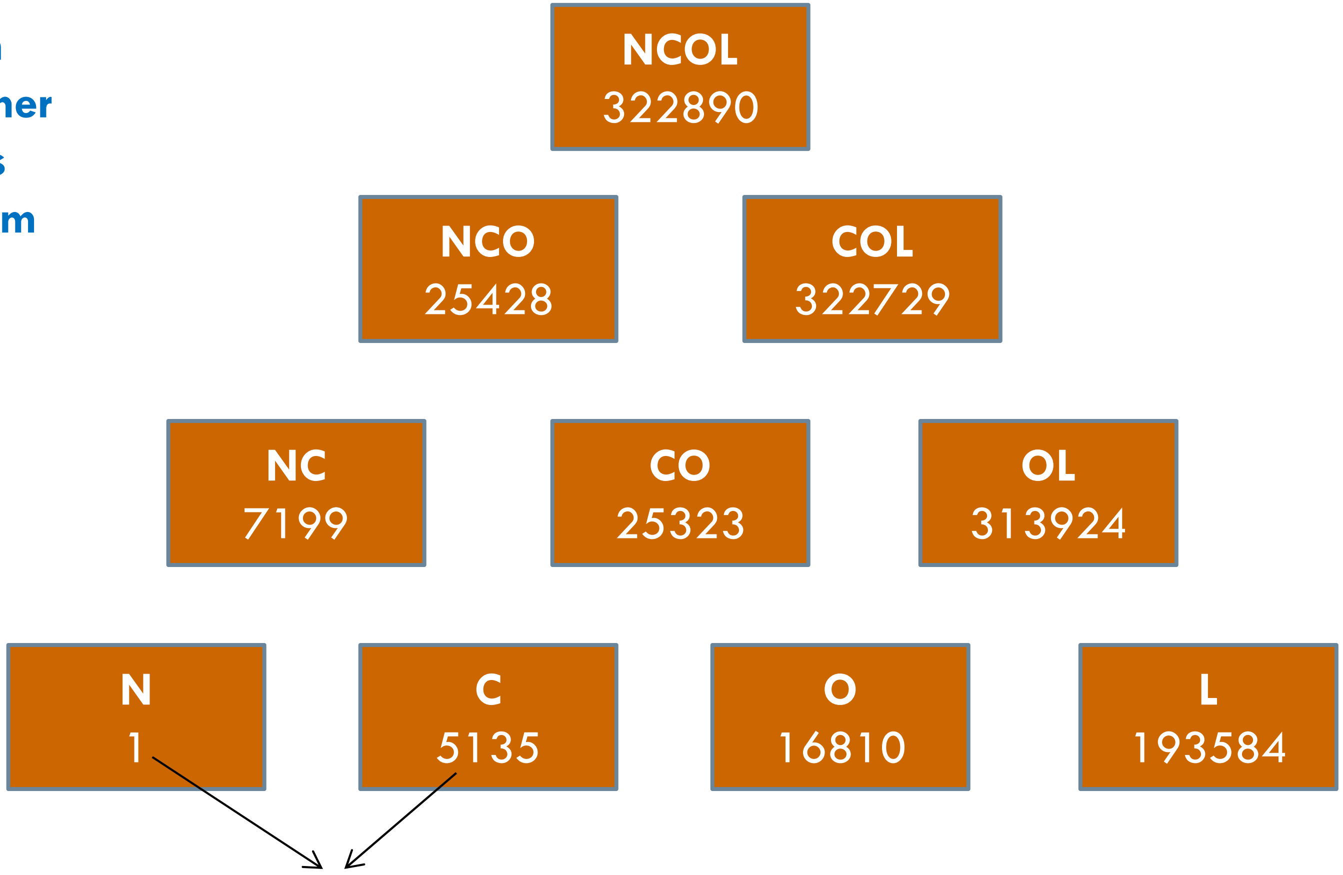
Fundamental Difficulty: Do not possess global knowledge about behavior in entire selectivity space!

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

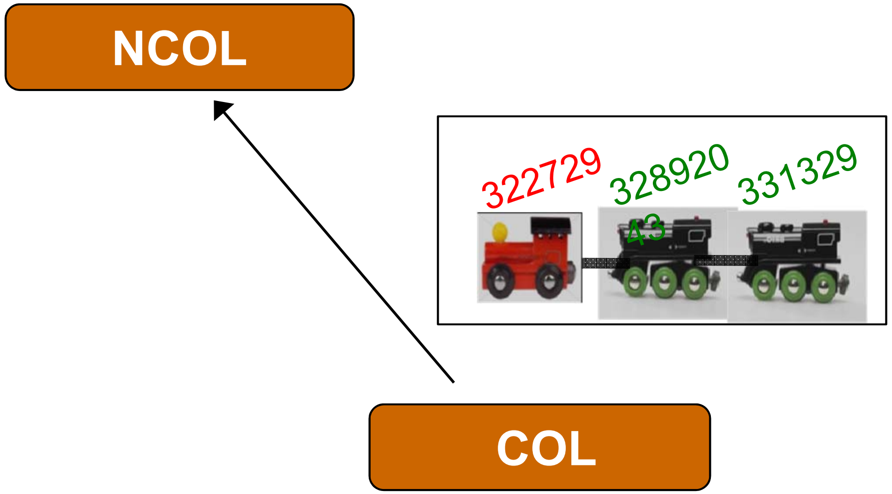
L
193584

Cost of the cheapest plan

EXPAND Plan Generation Algorithm

Plan Trains

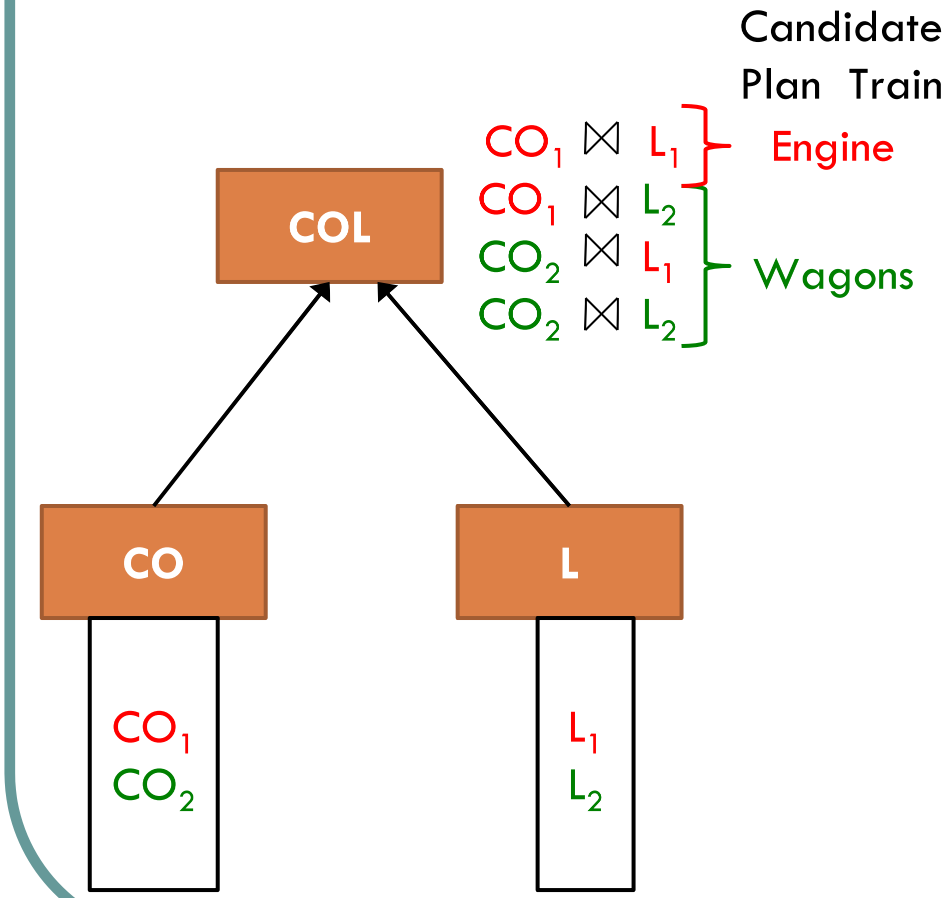
- At error-sensitive nodes of the DP-lattice, form a "plan train" that retains the cheapest plan ("engine") and, in addition, more expensive but stable candidates ("wagons")



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

Wagon Pruning

- 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]

Local Cost
322890
322901
324203
329089
329100
329229
334801
390748
395288

Check 1: Local Cost

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

- Eliminate all wagon sub-plans $\mathbf{p_w}$ with
  $$c(p_w, q_e) > (1 + \lambda)\, c(p_e, q_e)$$

After Local Cost Check (λ = 20%)

Local Cost
322890
322901
324203
329089
329100
329229
334801
~~390748~~
~~395288~~

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 (λ = 20%)

Local Cost	**V_0**	**V_1**	**V_2**	**V_3**
322890	202089	224599	846630	1271678
322901	202101	224610	846642	1271689
~~324203~~	~~202089~~	~~224604~~	~~846636~~	~~1952627~~
329089	208207	230766	356555	1280663
329100	208219	230777	356567	1280674
~~329229~~	~~202090~~	~~224928~~	~~846959~~	~~4563459~~
334801	214078	236628	362417	1204051

V_i: Four Corners of S

Check 3: Global Benefit

- Benefit Index (heuristic): Arithmetic Mean of corner costs

$$\xi(p_w, p_e) = \frac{\overline{c}(p_e, q_a)}{\overline{c}(p_w, q_a)} \qquad q_a \in Corners(\mathsf{S})$$

- 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

Local Cost	V_0	V_1	V_2	V_3	Benefit Index
322890	202089	224599	846630	1271678	1.0
~~322901~~	~~202101~~	~~224610~~	~~846642~~	~~1271689~~	~~0.99~~
329089	208207	230766	356555	1280663	1.22
329100	208219	230777	356567	1280674	1.22
334801	214078	236628	362417	1204051	1.26

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

Local Cost	V_0	V_1	V_2	V_3	Benefit Index
322890	202089	224599	846630	1271678	1.0
329089	208207	230766	356555	1280663	1.22
~~329100~~	~~208219~~	~~230777~~	~~356567~~	~~1280674~~	~~1.22~~
334801	214078	236628	362417	1204051	1.26

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

Big difference!

Local Cost	V_0	V_1	V_2	V_3	Benefit Index
322890	202089	224599	846630	1271678	1.0
329089	208207	230766	356555	1280663	1.22
334801	**214078**	**236628**	**362417**	**1204051**	**1.26** ✓

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

High replacement %

Error Immunity

Good Robustness

Good Help%

Que Temp late	EXPAND				SEER			
	REP %	Agg SERF	Max SERF	Help %	REP %	Agg SERF	Max SERF	Help %
QT5	85	0.54	1	55	47	0.61	1	64
QT10	98	0.21	1	20	37	0.21	1	20
3DQT8	69	0.18	1	10	59	0.17	1	9
3DQT10	99	0.39	1	44	24	0.38	1	41
DSQT7	93	0.28	1	28	46	0.28	1	28
DSQT26	30	0.49	1	50	29	0.49	1	49

- Performance comparable to SEER (global knowledge)!

2. Plan Diagram Characteristics

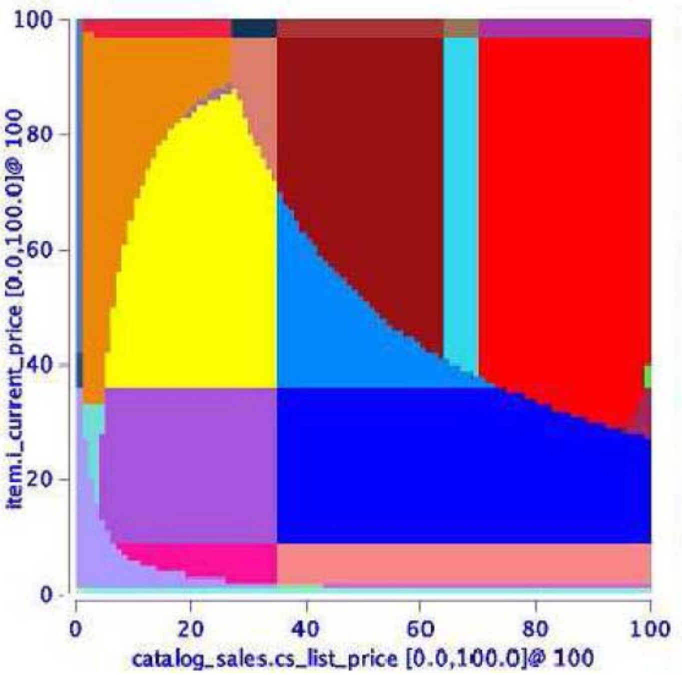
Anorexic diagrams

Query Template	OP P s	EXPAND Plans	 Non POSP	SEER Plans
QT5	11	3	0	2
QT10	15	3	0	2
3DQT8	43	3	0	2
3DQT10	30	5	1	3
DSQT7	12	2	1	2
DSQT26	13	2	1	2

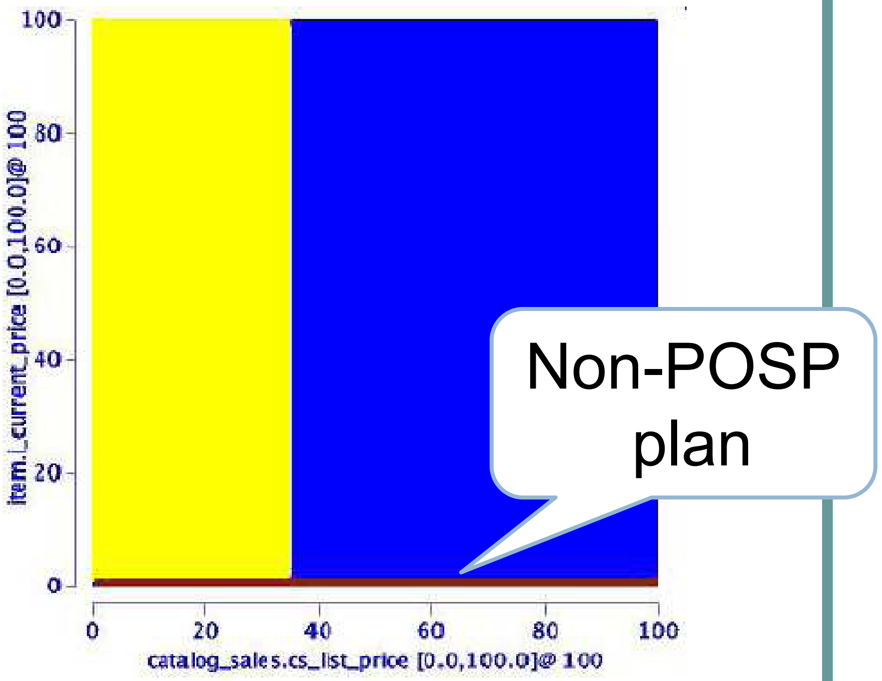
Non-POSP plans

Sample Plan Diagrams

[AIDSQT18]



DP: 28 plans



EXPAND: 3 plans

3. Time Overheads

Query Template	Optimization Time (ms) DP	 EXPAND
QT5	3.2	22.2 (+19.0)
QT10	0.9	3.2 (+2.3)
3DQT8	3.5	30.6 (+27.1)
3DQT10	0.9	4.3 (+3.4)
DSQT7	1.3	7.7 (+6.4)
DSQT26	1.4	7.0 (+5.6)

- Additional time of < 100ms
  - Miniscule compared to expected execution time savings

4. Memory Overheads

Query Template	Memory Overheads (MB) DP	 EXPAND
QT5	2.8	7.0 (+4.2)
QT10	2.2	3.4 (+1.2)
3DQT8	4.0	10.6 (+6.6)
3DQT10	2.2	5.1 (+2.9)
DSQT7	2.4	3.5 (+1.1)
DSQT26	2.4	3.8 (+1.4)

- Extra memory of < 100 MB
- Held very briefly (<< 100 ms)

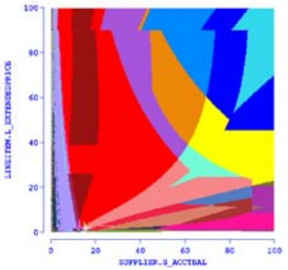
Summary

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
- acceptable overheads.

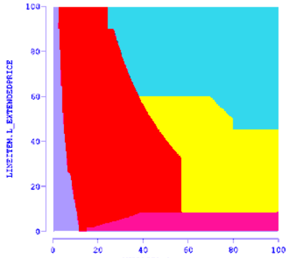
Take Away

- PICASSO → **[VLDB05]**



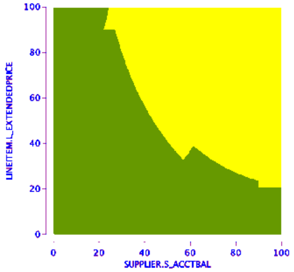
- Dense and Intricate Plan Diagrams
- PQO violations
- Optimizer bugs

- Cost Greedy → **[VLDB07]**



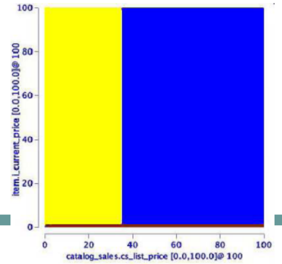
- Anorexic Reduction (less than10 plans)
- Local Near-optimality (20%)

- SEER / GSPQO → **[VLDB08]**



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

- EXPAND → **[VLDB10]**



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

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.

3. Plan Reduction Theory

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.

5. Query Execution Visualization

Plan diagrams capture the “compile-time” behavior of query optimizers. Useful to also visualize the run-time behavior in a similar manner [CIDR2009]

PRIMARY REFERENCES

- "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.

More Details:

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

END TUTORIAL