# Picasso: Drawing Out the Artistic Talents of DB Query Optimizers

Jayant Haritsa
Database Systems Lab
Indian Institute of Science

# Prof Jayant Haritsa

Prof. Jayant R. Haritsa is on the distinguished faculty of the Supercomputer Education & Research Centre and the Department of Computer Science & Automation at the Indian Institute of Science, Bangalore, since 1992.

He received the BTech degree in Electronics and Communications Engineering from the Indian Institute of Technology (Madras), and the MS and PhD degrees
in Computer Science from the University of Wisconsin (Madison).

He is on the editorial board of numerous International Publications. He is a Member of the Mathematical Sciences Research Committee, CSIR, Govt. of India. He is also a Distinguished Scientist of ACM, Senior Member of IEEE and Life Member of CSI.

His honours include:

- ✓ Distinguished Alumnus, Indian Institute of Technology (Madras), India, 2012
- ✓ Fellow, Indian Academy of Sciences (IASc), India, 2010
- ✓ Fellow, Indian National Academy of Engineering (INAE), India, 2009
- ✓ Fellow, The National Academy of Sciences (NASI), India, 2006

- ✓ Shanti Swarup Bhatnagar Prize, CSIR, 2009
- ✓ Hari Om Ashram Prerit Dr. Vikram Sarabhai Research Award, PRL, 2007
- ✓ Swarnajayanti Fellowship, Dept. of Science & Technology, Govt. of India, 2002
- ✓ Sir C V Raman Young Scientist Award in Computer Science, Govt. of Karnataka, 2001
- ✓ Research Awards from IBM, Google and Microsoft

His research interests are in database systems.

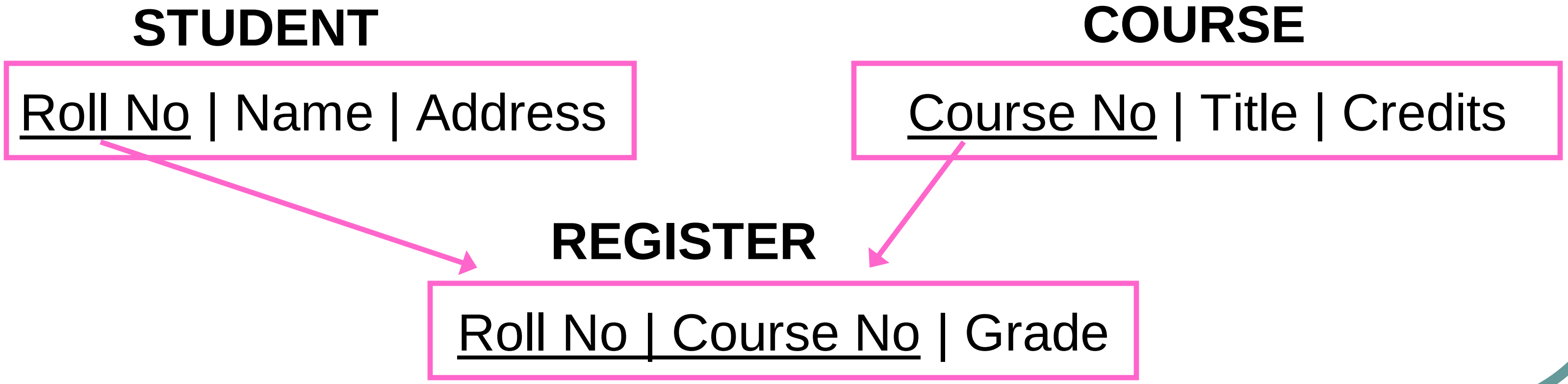
His Research focus and publications can be found here:
http://dsl.serc.iisc.ernet.in/~haritsa/

# Prof Jayant Haritsa

# Relational Database Systems
[RDBMS]

- Based on first-order logic
  - Edgar Codd of IBM Research, Turing Award (1981)
  - “We believe in Codd, not God”

- Data is stored in a set of relations (i.e. tables) with attributes, relationships, constraints



**STUDENT**

Roll No | Name | Address

**COURSE**

Course No | Title | Credits

**REGISTER**

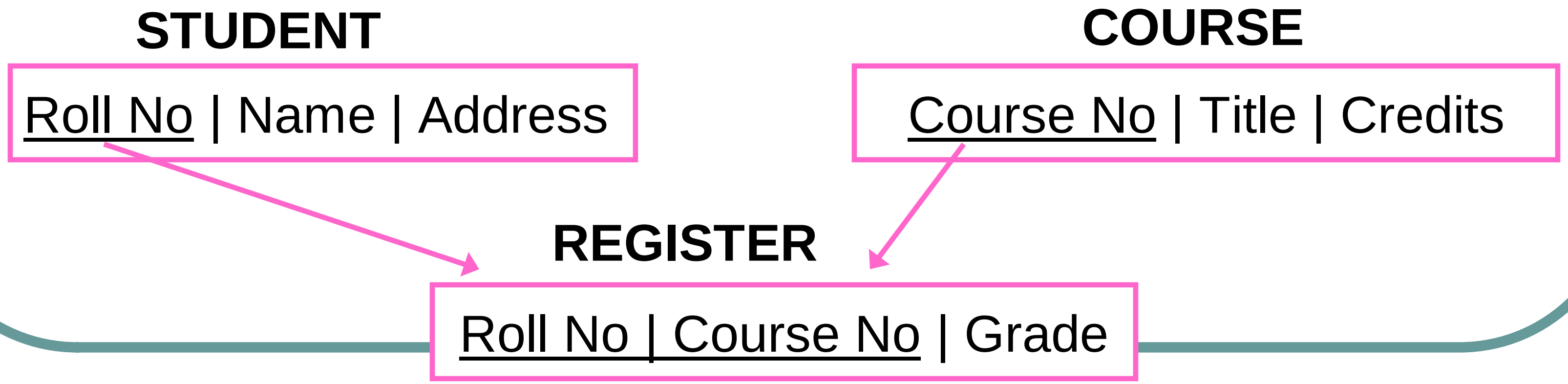
Roll No | Course No | Grade

# QUERY INTERFACE

## Structured Query Language (SQL)

- Invented by IBM, 1970s
- Example: *List names of students and their course titles*

```
select  STUDENT.Name, COURSE.Title
 from    STUDENT,  COURSE,  REGISTER
where  STUDENT.RollNo = REGISTER.RollNo   and
           REGISTER.CourseNo = COURSE.CourseNo
```



# Query Execution Plans

- SQL is a declarative language
  - Specifies only what is wanted, but not how the query should be evaluated (i.e. ends, not means)

```
select   STUDENT.Name, COURSE.Title
from     STUDENT,  COURSE,  REGISTER
where    STUDENT.RollNo = REGISTER.RollNo   and
            REGISTER.CourseNo = COURSE.CourseNo
```

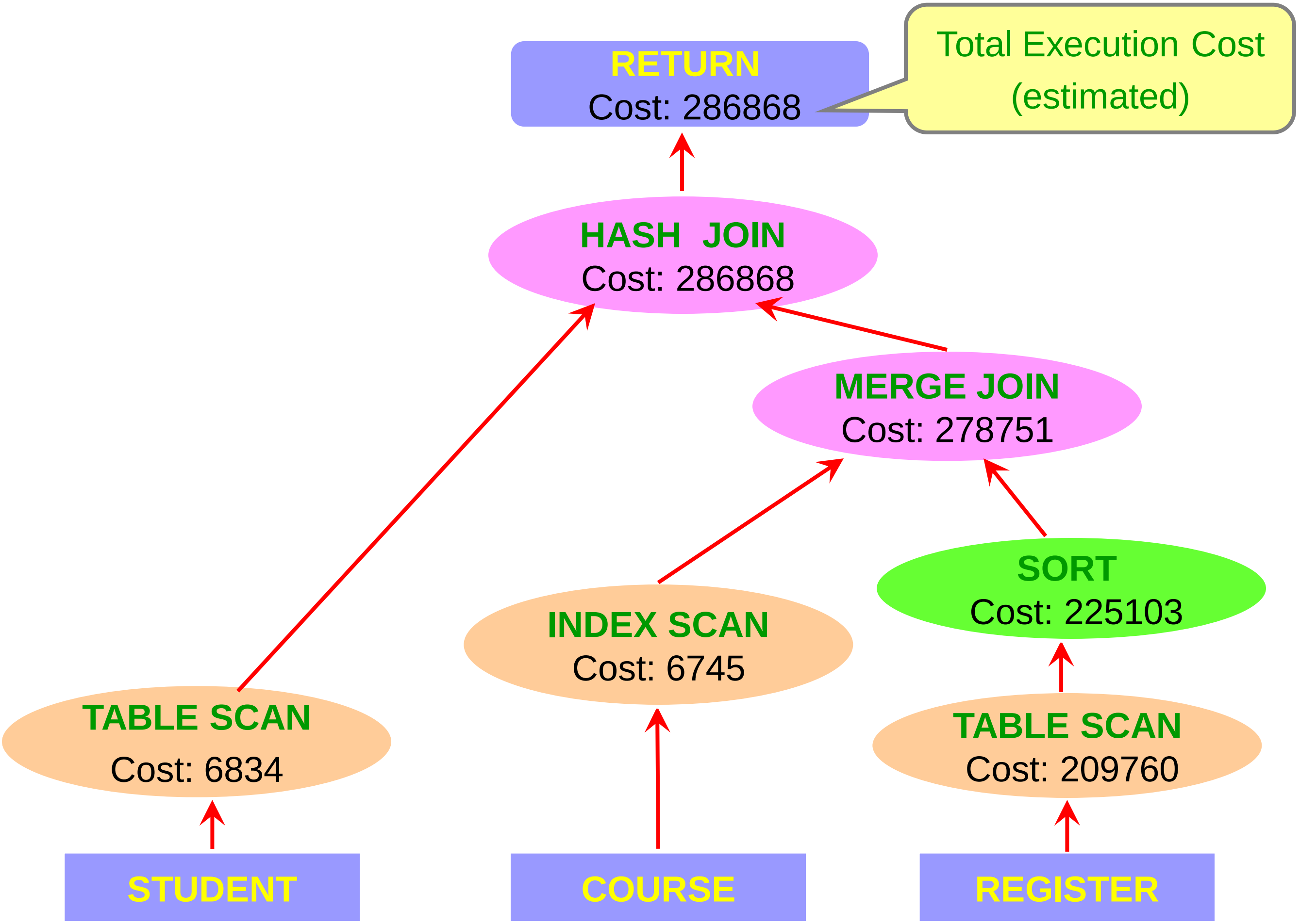
Unspecified:

**join order [ $((S \bowtie R) \bowtie C)$ or $((R \bowtie C) \bowtie S)$ ? ]**

**join techniques [ Nested-Loops or Sort-Merge or Hash ? ]**

- DBMS query optimizer identifies the optimal evaluation strategy: “query execution plan”

# Sample Execution Plan



**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**

# Query Plan Selection

- Core technique

SQL Query (Q)

Query Optimizer
(dynamic programming)

Minimum Cost
Plan P(Q)

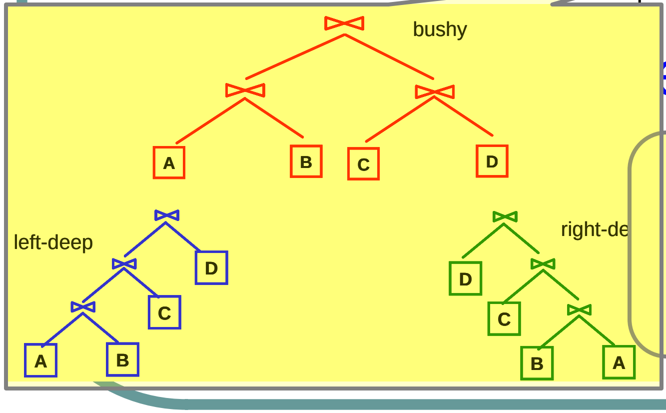
Search
Space

Cost
Model

DB
catalogs

Number of blocks
of relation R;
Number of unique
values of attribute A;
…

expen e since exhau



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

# Need for careful plan selection

- Cost difference between best plan choice and a random choice can be enormous (orders of magnitude!)
- Only a small percentage of really good plans over the (exponential) search space

# Relational Selectivities

- Cost-based Query Optimizer’s choice of

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

- For a given database and system setup,

  execution plan = 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 [Q7 of TPC-H]

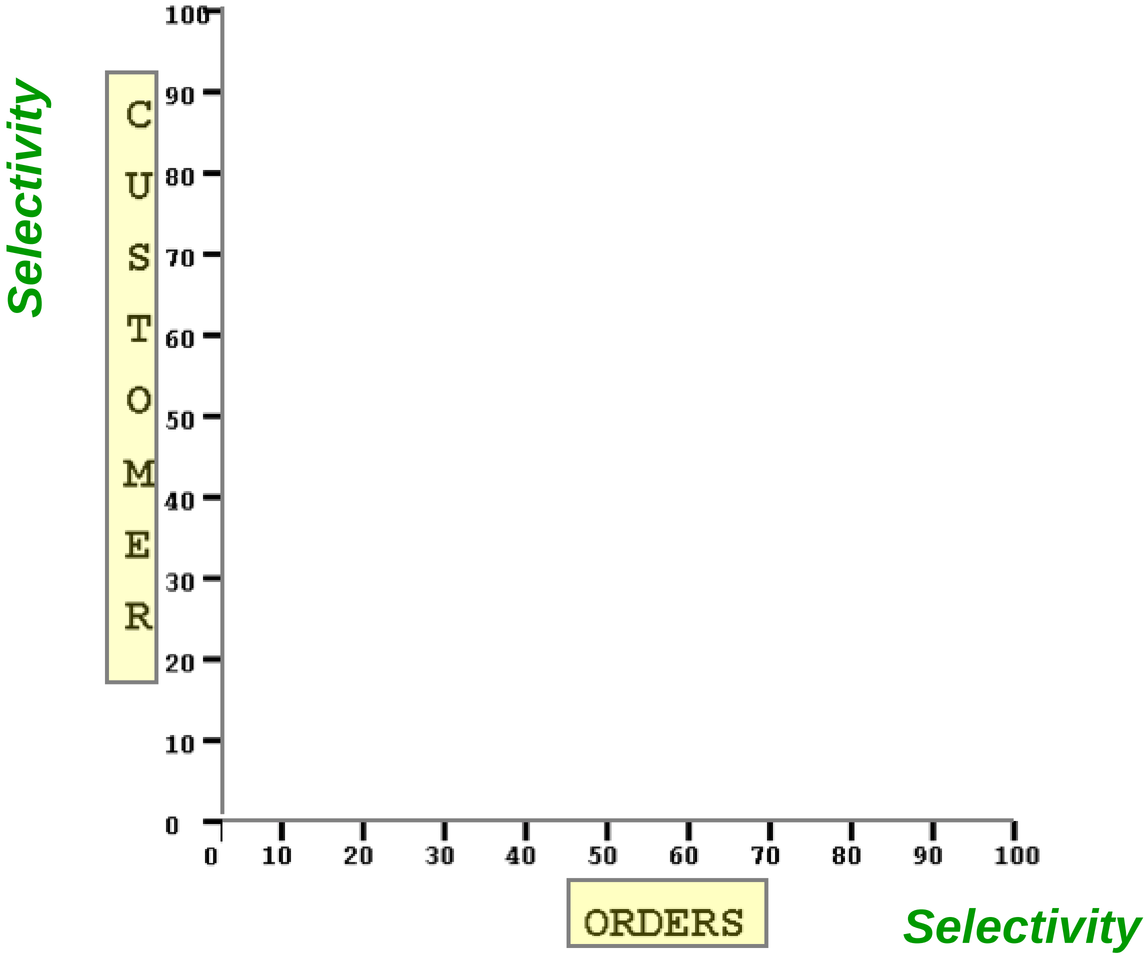
***Determines the values of goods shipped between nations in a time period***

```
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                      pkey and o_orderkey = l
                           key and s_nationkey = n
                          n2.n_nationkey and
        ((n1.n_name =   RANCE' and n2.n_name = 'GE      NY') or
         (n1.n_name = 'G   RMANY' and n2.n_name = '   ANCE')) and
         l_shipdate between   ate '1995-01-01' and da   e '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



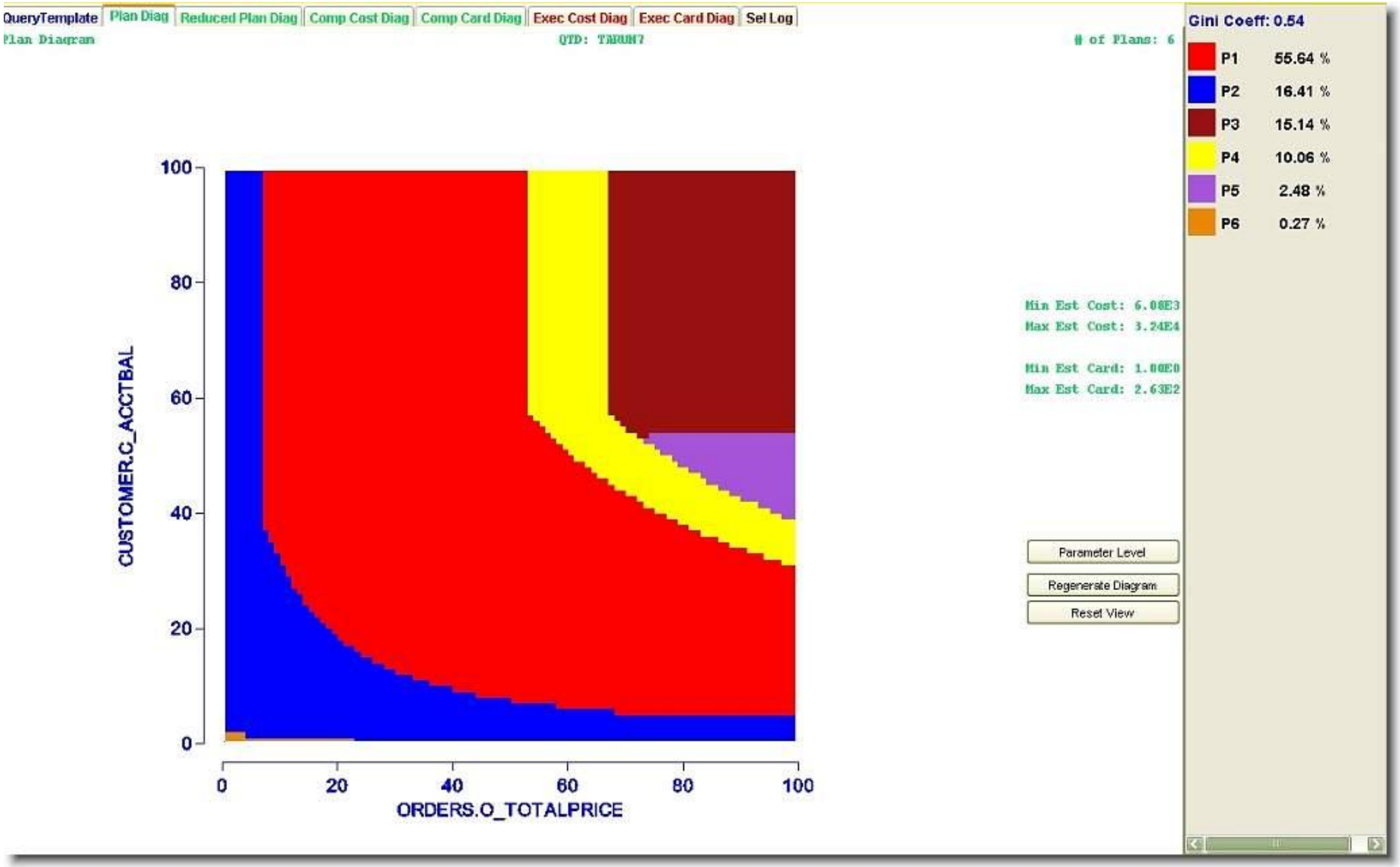
# Plan and Cost Diagrams

- 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]



Plan Diagram

QTD: TARUN7

# of Plans: 6

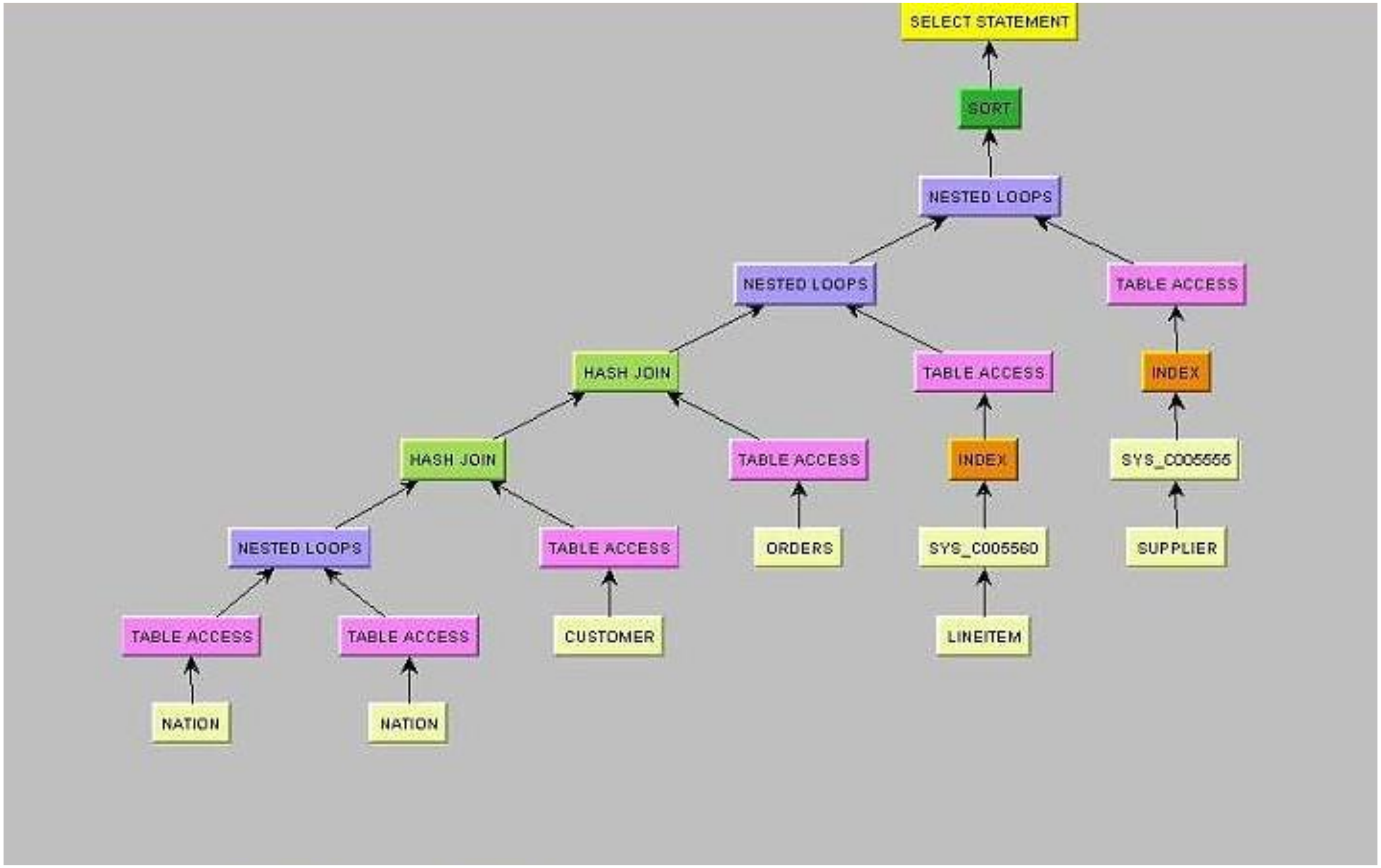
Gini Coeff: 0.54

| | | |
|---|---|---|
| 🟥 | 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

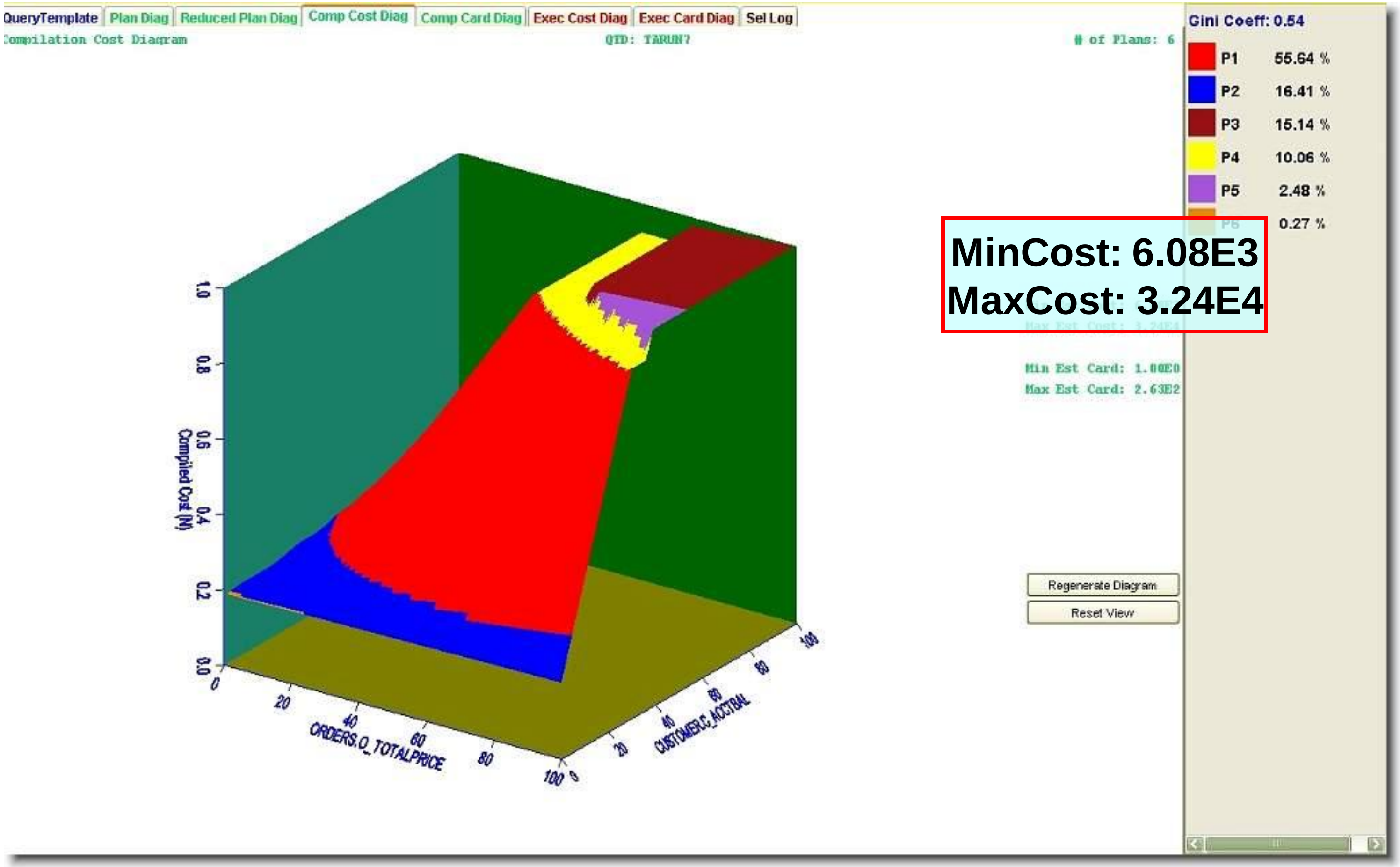
Min Est Card: 1.00E0
Max Est Card: 2.63E2

# Plan P13


SELECT STATEMENT
SORT
NESTED LOOPS
NESTED LOOPS
TABLE ACCESS
HASH JOIN
TABLE ACCESS
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]



Compilation Cost Diagram

QTD: TARUN7

# 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

# Part I:  PICASSO

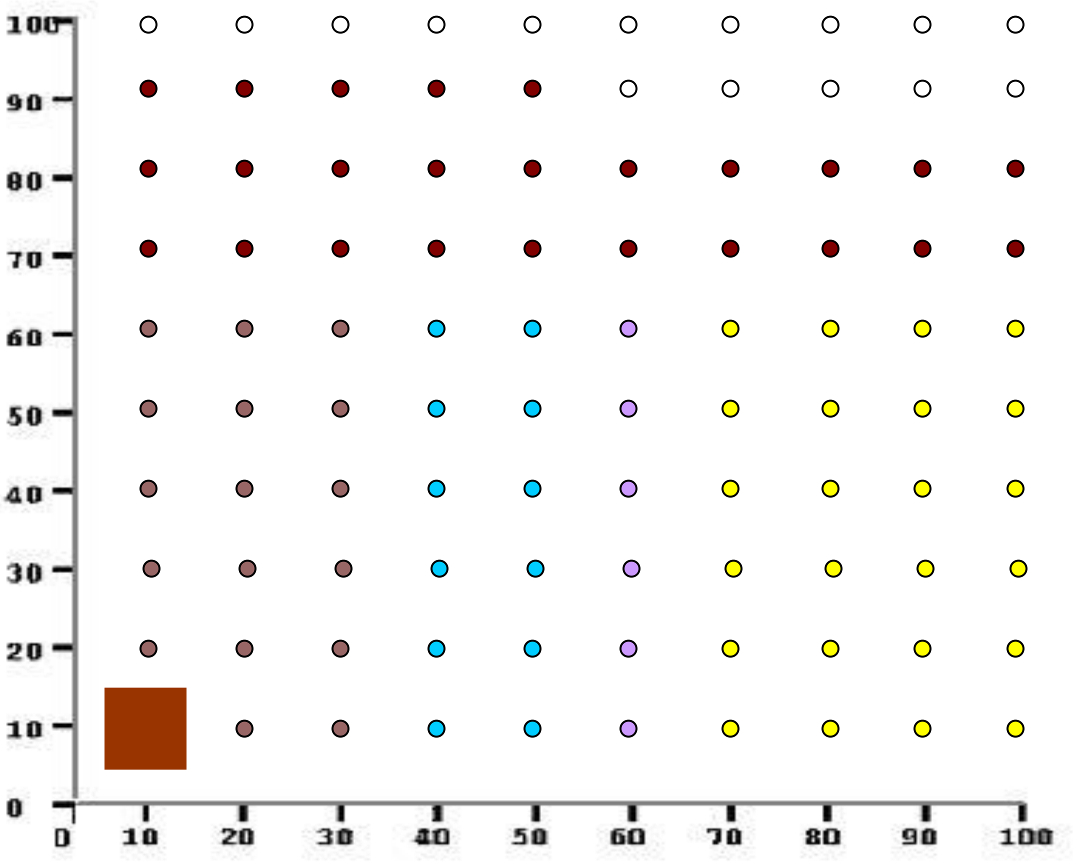
# Overview

Picasso is a Java tool that, given a multi-dimensional SQL query template and a choice of database engine, automatically generates plan, cost and card diagrams

- Fires queries at user-specified granularity (10, 30, 100, 300, 1000 queries per dimension)
- Visualization: 2-D plan diagrams (slices if n > 2)
  3-D cost and card diagrams

Also:
Plan-trees, Plan differences
Foreign Plans
Abstract-plan diagrams
Execution cost/card diagrams

# Diagram Generation Process



**CUSTOMER.**
**c_acctbal**

**ORDERS.o_totalprice**

# Tool Status

- ~50000 lines of code (2004-09) with ~100 classes
- Uses Java3D, VisAd, JGraph, Swing
- Operational on DB2/Oracle/SQLServer/Sybase/PostgreSQL
- Copyrighted by IISc in May 2006
- Released as free software in Nov 2006 by Associate Director of IISc
- Release of version 1.0 in May 2007,
  version 2.0 in Feb 2009, version 3.0 in April 2012
- In use at academic and industrial labs worldwide
  - CMU, Purdue, Duke, TU Munich, NU Singapore, IIT-B, …
  - IBM, Microsoft, Oracle, Sybase, HP, …
- Received Best Software award in
  Very Large Data Base (VLDB) conference, 2010

# Why do they care?

Excited the interest of industrial and academic communities

- serious problems and anomalies in current optimizer design
  - optimizer evaluator / debugger / designer
  - database administrators – response time fault profiler
- testbed for database researchers
- educational aid for students

# Why do we care?

- NOT s/w development for its own sake !
- Development of the tool has thrown up many interesting core CS research problems involving theory, algorithms, statistics, tree matching, …

# PICASSO OUTPUT

Full result listing at http://dsl.serc.iisc.ernet.in/projects/PICASSO

# Testbed Environment

- Databases
  - TPC-H database (1 GB)
  - TPC-DS database (100 GB)

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

- Query Sets
  - 2-D, 3-D, 4-D Query templates based on TPC-H benchmark [Q1 ~ Q22] and TPC-DS benchmark [Q1 ~ Q99]
  - Default uniform 100x100 grid (10000 queries) [0.5%, 0.5%] to [99.5%, 99.5%]
- Relational Engines
  - Default installations (with all optimization features on)
  - Stats on all columns; no extra indices
- Computational Platforms
  - PIV 2.4 GHz, 2GB RAM, Windows XP Pro
  - Sun Opteron 4GHz, 4GB RAM, Windows XP Pro

# The Picasso Connection

Plan diagrams are often similar to cubist paintings !

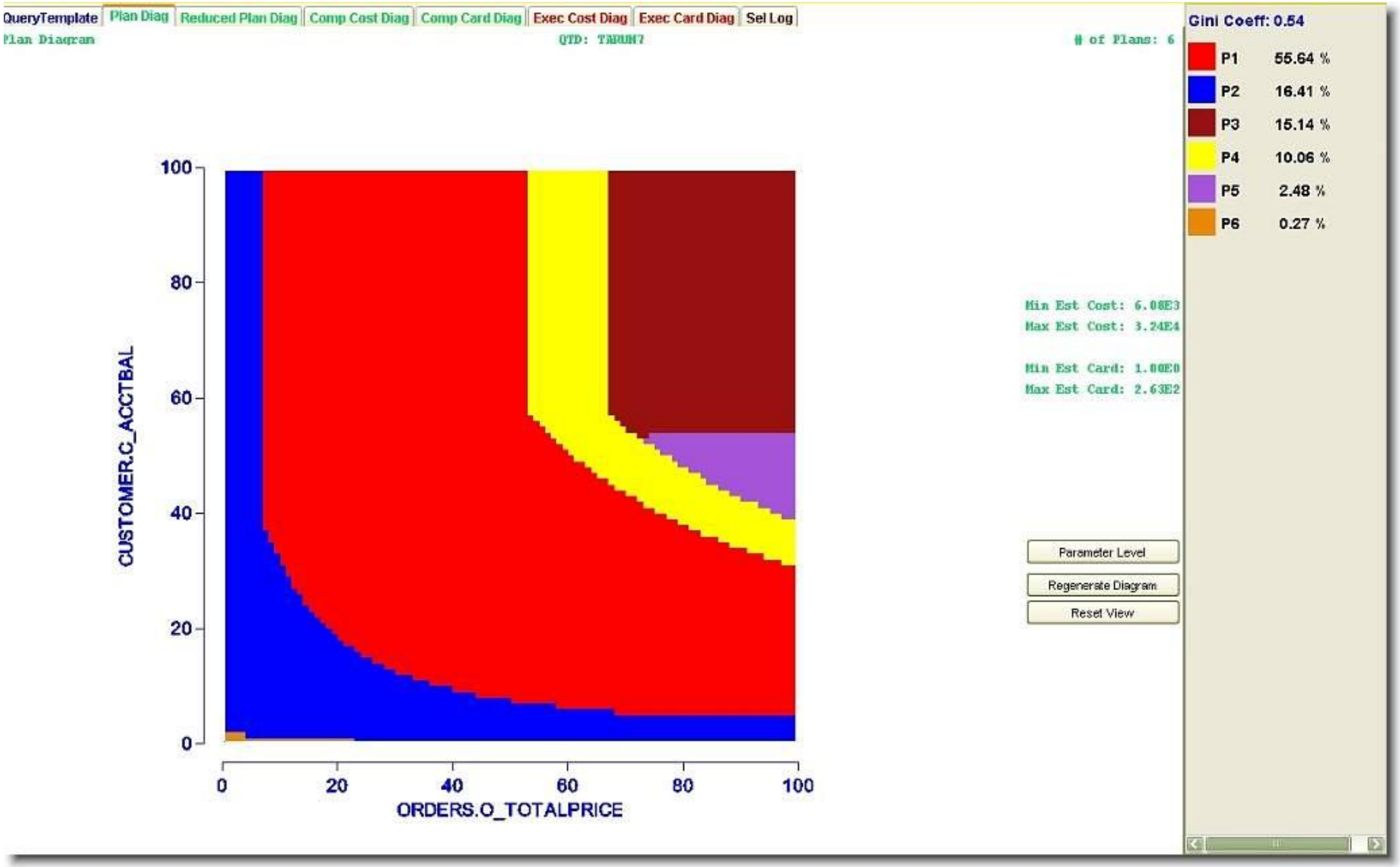
[ Pablo Picasso – founder of cubist genre ]

***Woman with a guitar***

Georges Braque, 1913

# Smooth Plan Diagram

## [QT7,OptB]



Plan Diagram

QTD: TARUN7

# 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: 6.00E3
Max Est Cost: 3.24E4

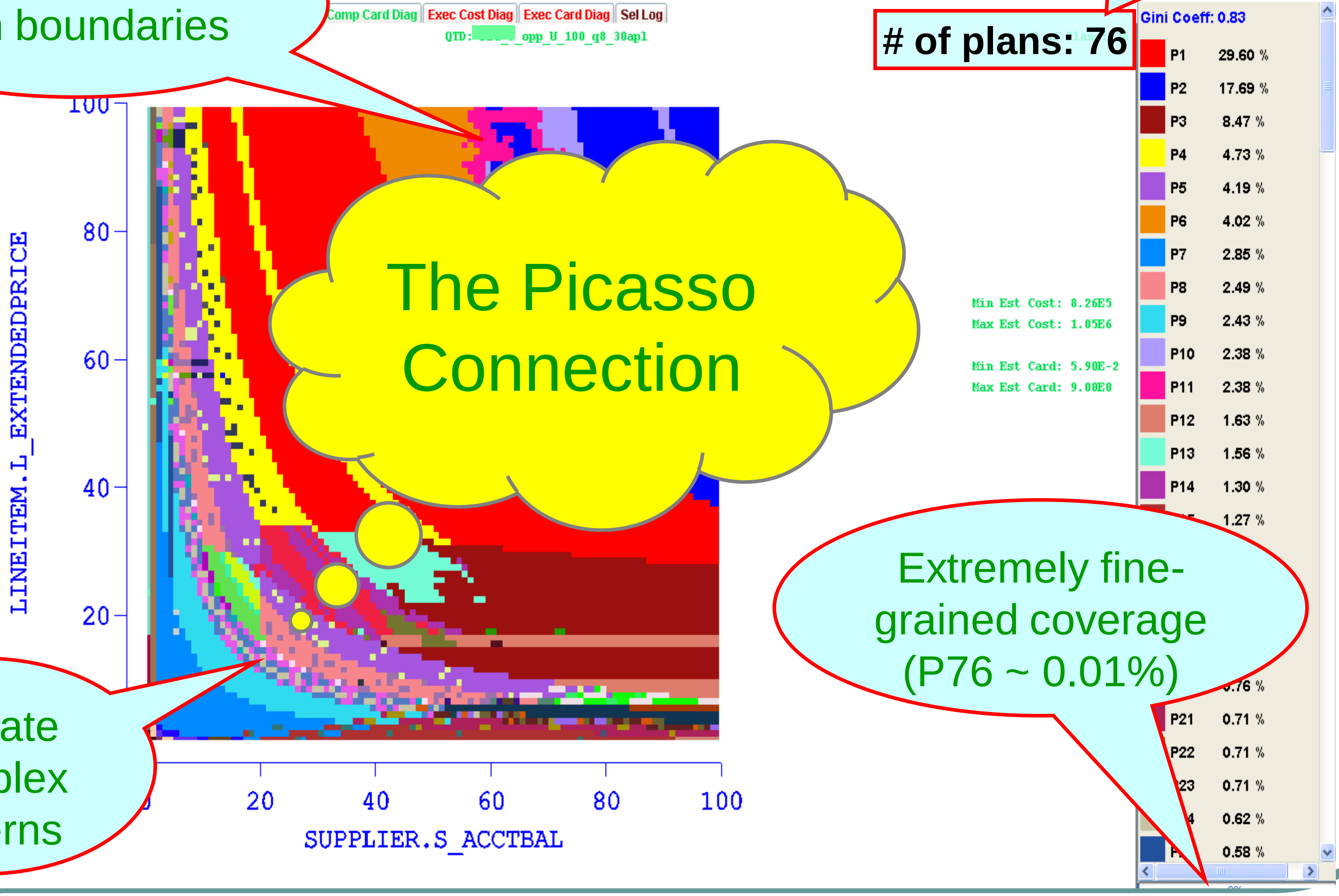
Min Est Card: 1.00E0
Max Est Card: 2.63E2

# Complex Plan Diagram

## [QT8.OptA*]

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

Highly irregular plan boundaries



QTD:[redacted]_opp_U_100_q8_30apl

**# of plans: 76**

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

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

Gini Coeff: 0.83

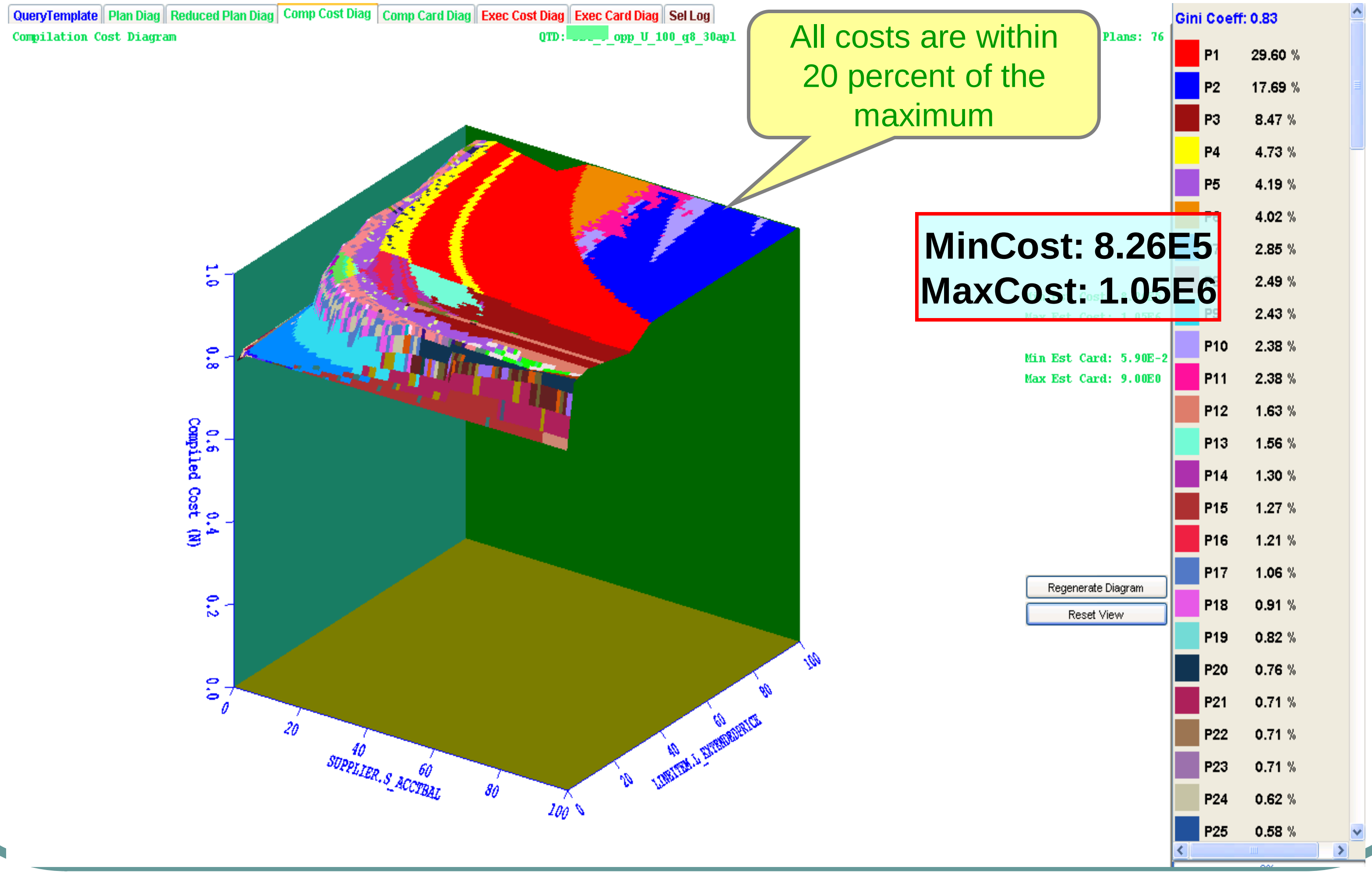
| | | |
|---|---|---|
| 🟥 | P1 | 29.60 % |
| 🟦 | P2 | 17.69 % |
| 🟫 | P3 | 8.47 % |
| 🟨 | P4 | 4.73 % |
| 🟪 | P5 | 4.19 % |
| 🟧 | P6 | 4.02 % |
| 🟦 | P7 | 2.85 % |
| 🟥 | P8 | 2.49 % |
| 🟦 | P9 | 2.43 % |
| 🟪 | P10 | 2.38 % |
| 🟥 | P11 | 2.38 % |
| 🟫 | P12 | 1.63 % |
| 🟩 | P13 | 1.56 % |
| 🟪 | P14 | 1.30 % |
| | | 1.27 % |
| | | .76 % |
| | P21 | 0.71 % |
| | P22 | 0.71 % |
| | 23 | 0.71 % |
| | 4 | 0.62 % |
| | P | 0.58 % |

The Picasso Connection

Extremely fine-grained coverage (P76 ~ 0.01%)

Intricate Complex Patterns

# Cost Diagram
## [QT8, Opt A*]



QTD: [redacted]_opp_U_100_q8_30apl

Plans: 76

All costs are within 20 percent 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 % |
| P6 | 4.02 % |
| | 2.85 % |
| | 2.49 % |
| P9 | 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 % |

# Remarks

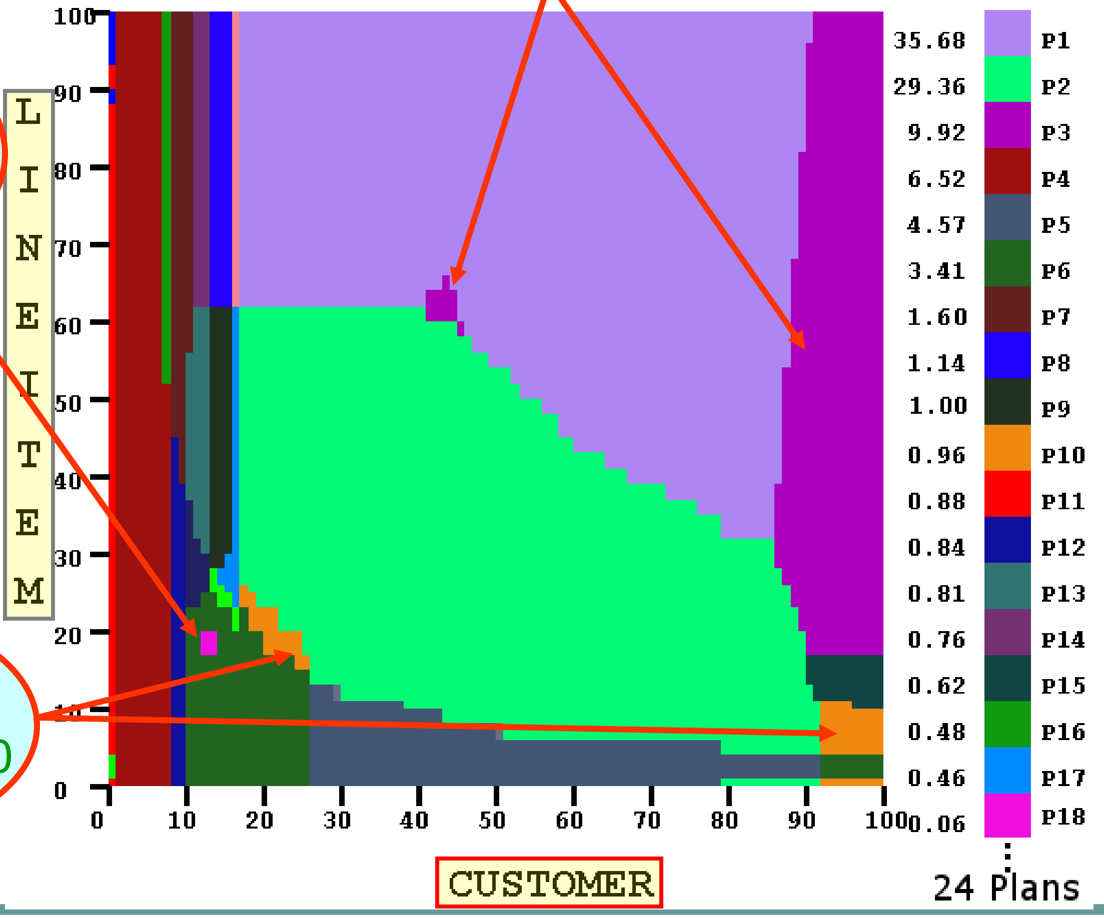
- Modern optimizers tend to make extremely fine-grained and skewed choices
- Is this an over-kill, perhaps not merited by the coarseness of the underlying cost space – i.e. are optimizers “doing too good a job” ?
- Is it feasible to reduce the plan diagram complexity without materially affecting the plan quality? (Part II)

# Picasso Art Gallery

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

# Duplicates and Islands

[QT10, OptA]



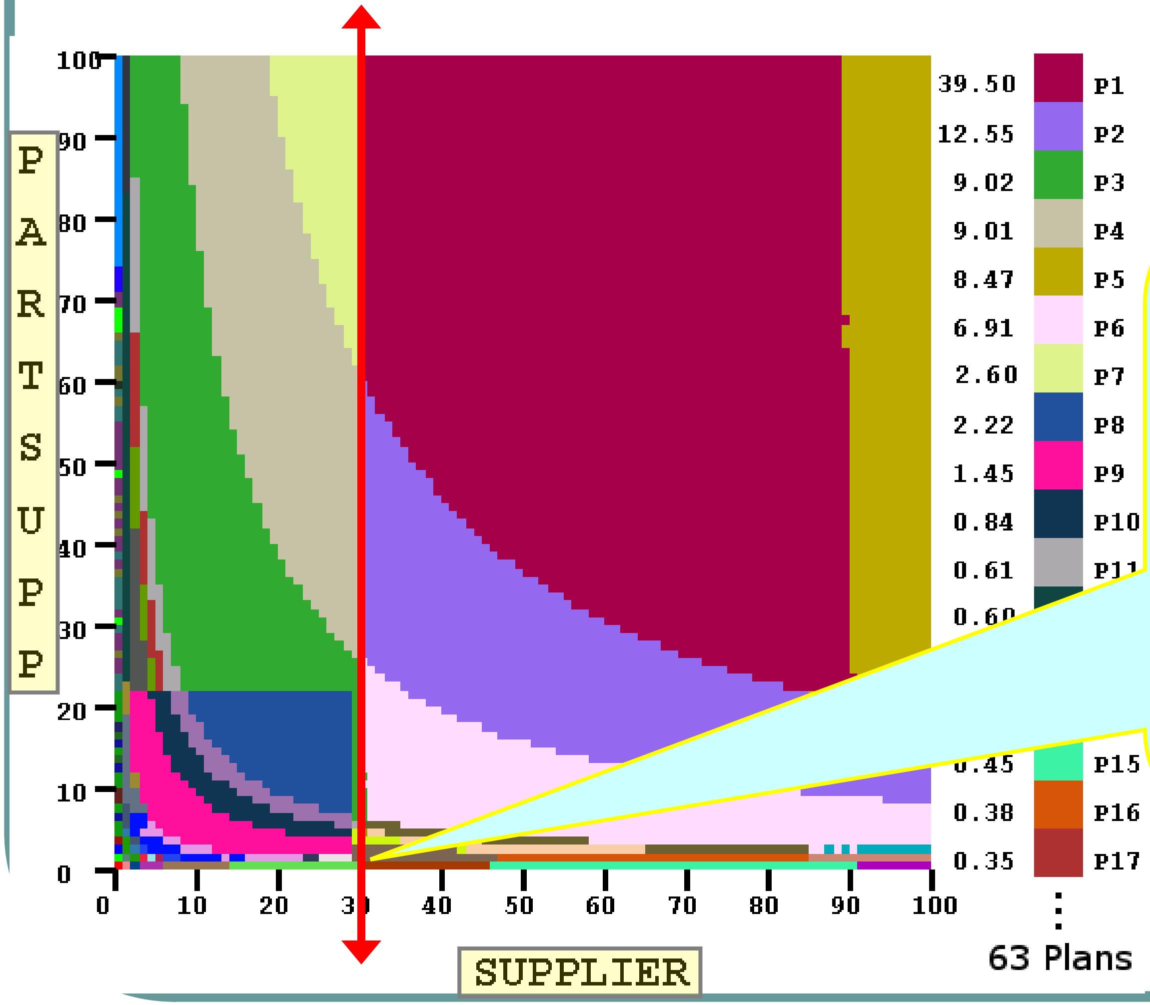
Duplicate locations of P3

P18 is an island within P6

Duplicate locations of P10

# Plan Switch Points

## [QT9,OptA]

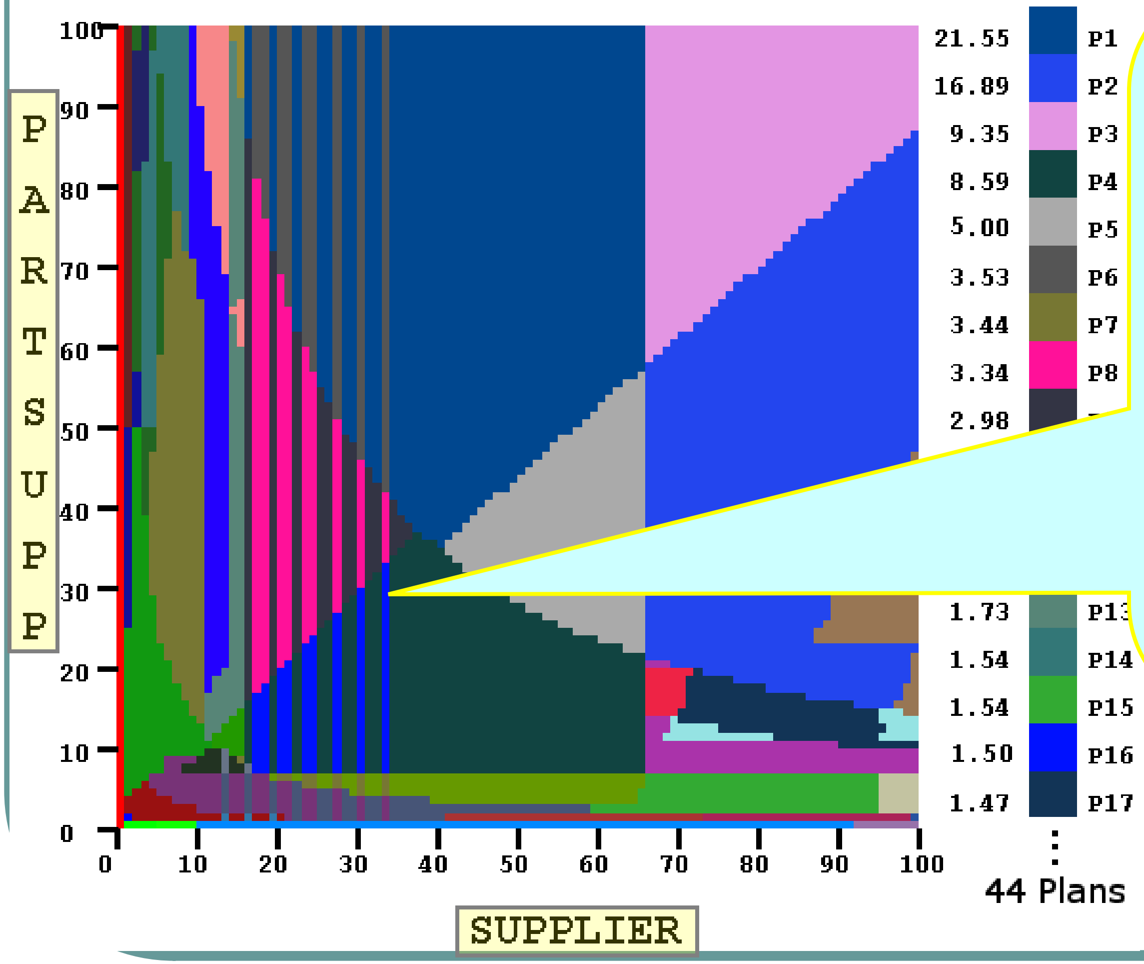


Plan Switch Point:
line parallel to axis with a plan shift for all plans bordering the line.

Hash-Join sequence
PARTSUPP▶◀SUPPLIER▶◀PART
is altered to
PARTSUPP▶◀PART▶◀SUPPLIER

# Venetian Blinds

## [QT9,OptB]

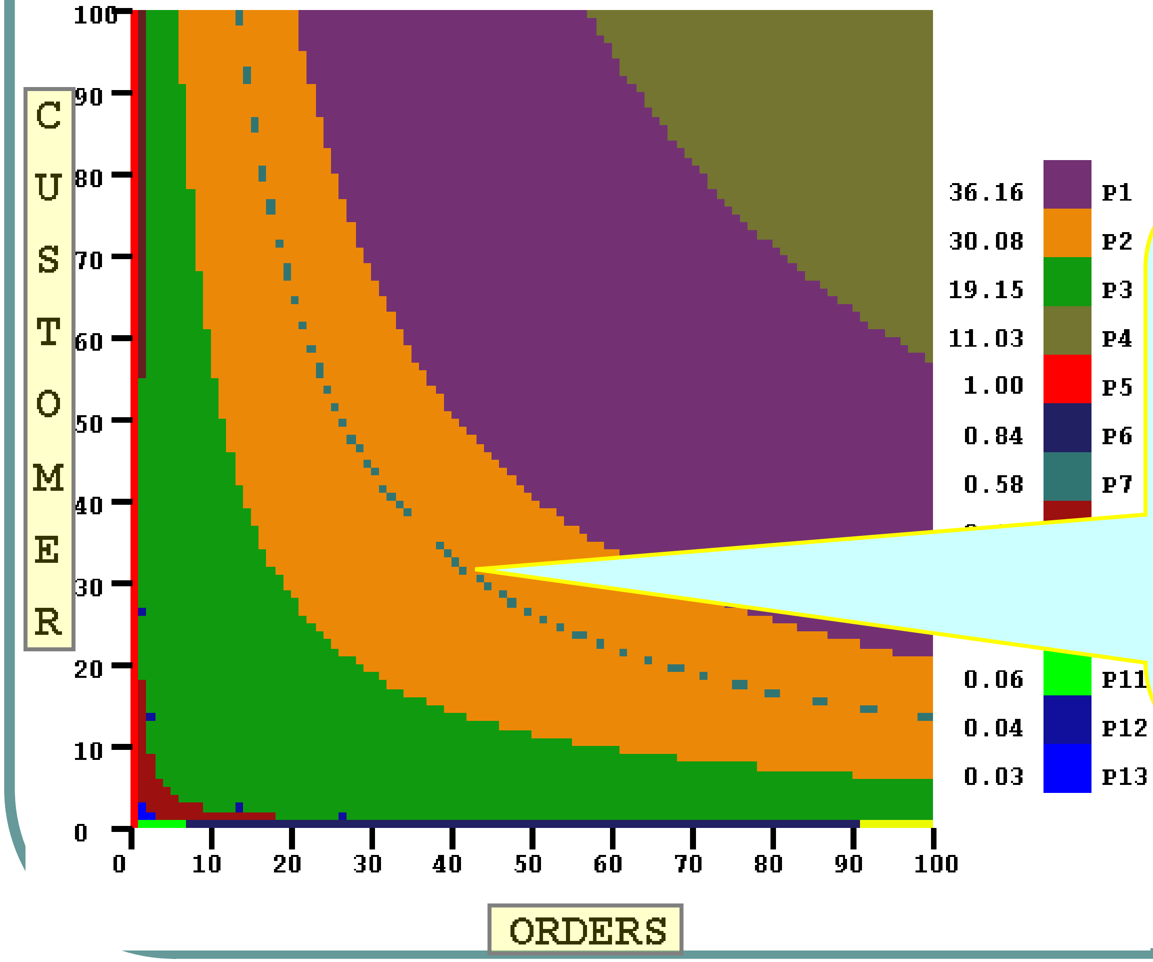


Six plans simultaneously change with rapid alternations to produce a "Venetian blinds" effect.

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

# Footprint Pattern

## [QT7,OptA]

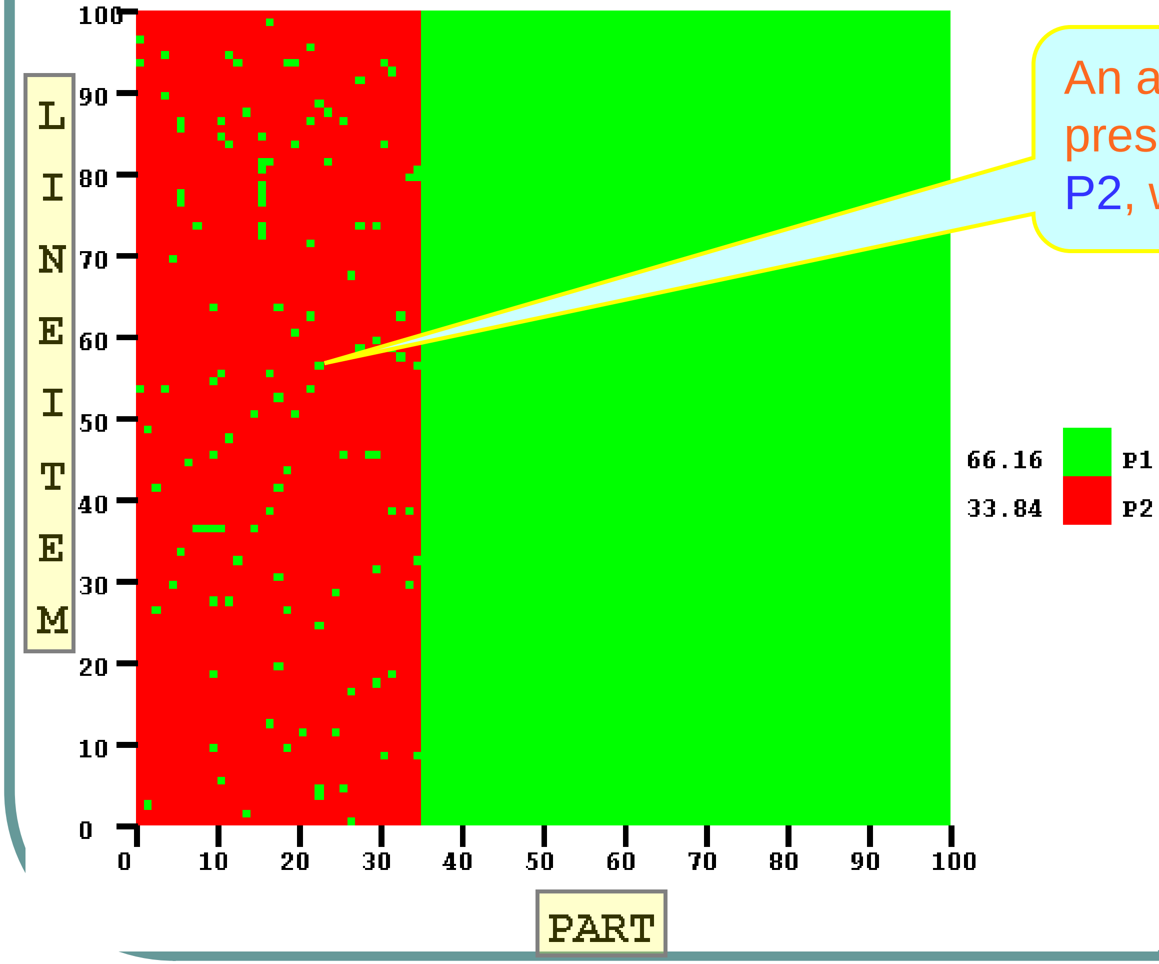


P7 is a thin and broken curved pattern in the middle of P2's region.

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

# Speckle Pattern

## [QT17,OptA]



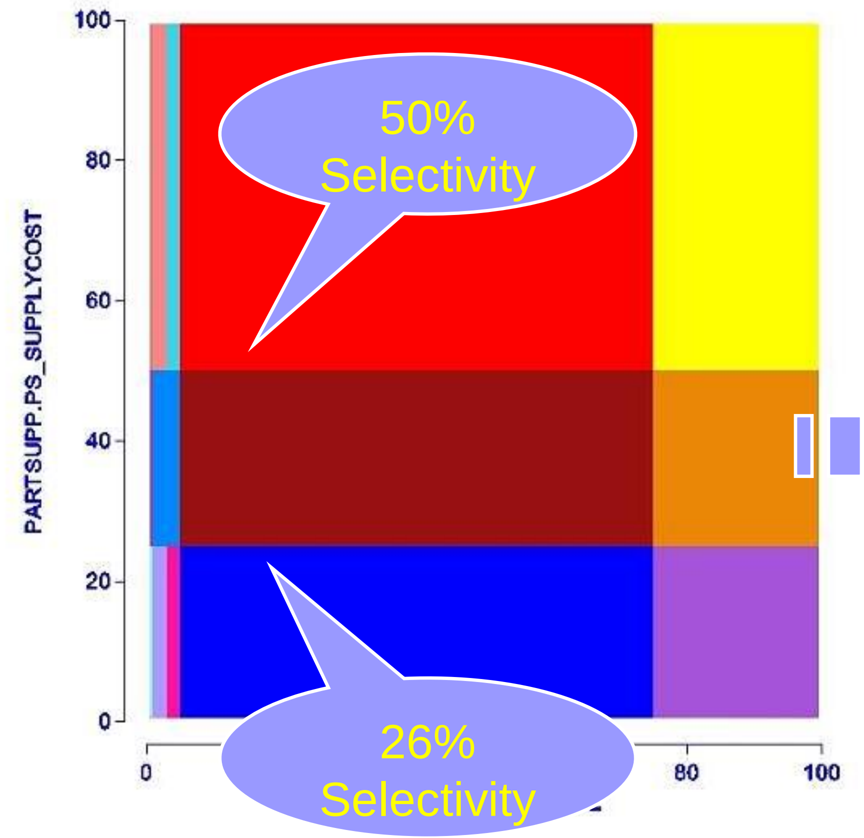
An additional sort operation is present on PART relation in P2, whose cost is very low

# Non-Monotonic Cost Behavior

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

# Plan-Switch Non-Monotonic Costs

## [QT2,OptA]



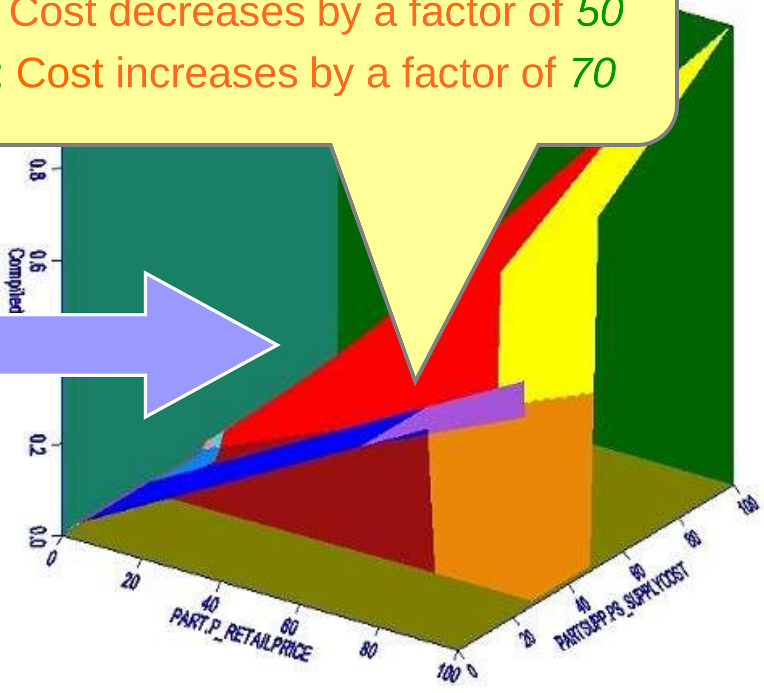
50% Selectivity

26% Selectivity

Plan Diagram

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

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

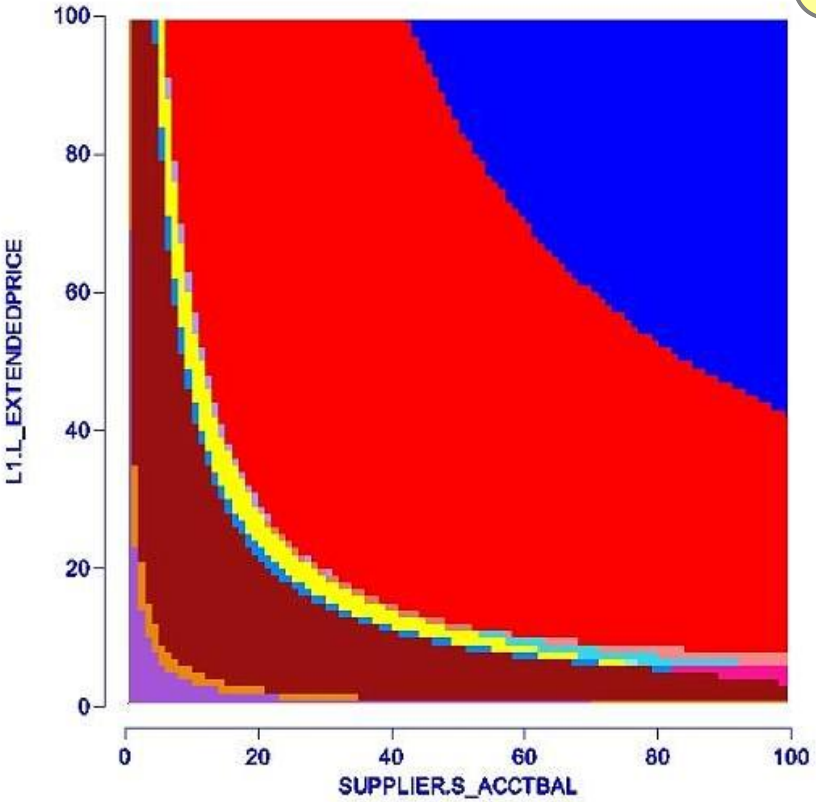


Cost Diagram

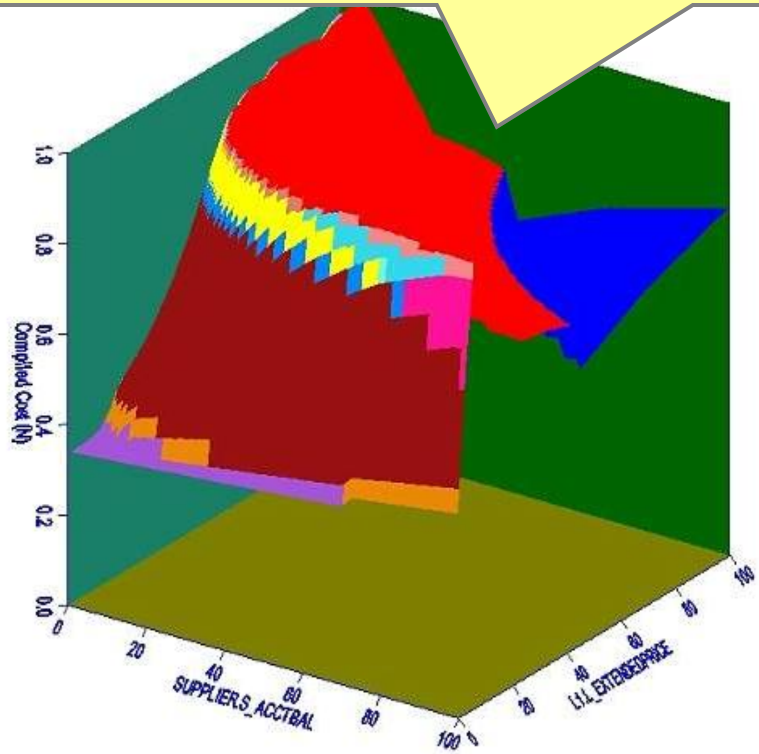
# Intra-Plan Non-Monotonic Costs

[QT21,OptA]

*Nested loops join whose cost decreases with increasing input cardinalities*



Plan Diagram



Cost Diagram

# Remarks

- Optimizers may have become too complex over time, making it difficult to anticipate the interactions and side-effects of their modules
- Well-kept secret by optimizer developers? Perhaps worth having a re-look at optimizer design …

# Part II: PLAN DIAGRAM REDUCTION

# 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 $\lambda$ percent* (user-defined)

Analogy:

(with due apologies to Sri Lankans in the audience)
Sri Lanka agrees to be annexed by India if it is assured that the cost of living of each Lankan citizen is not increased by more than $\lambda$ percent

# Our Results

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

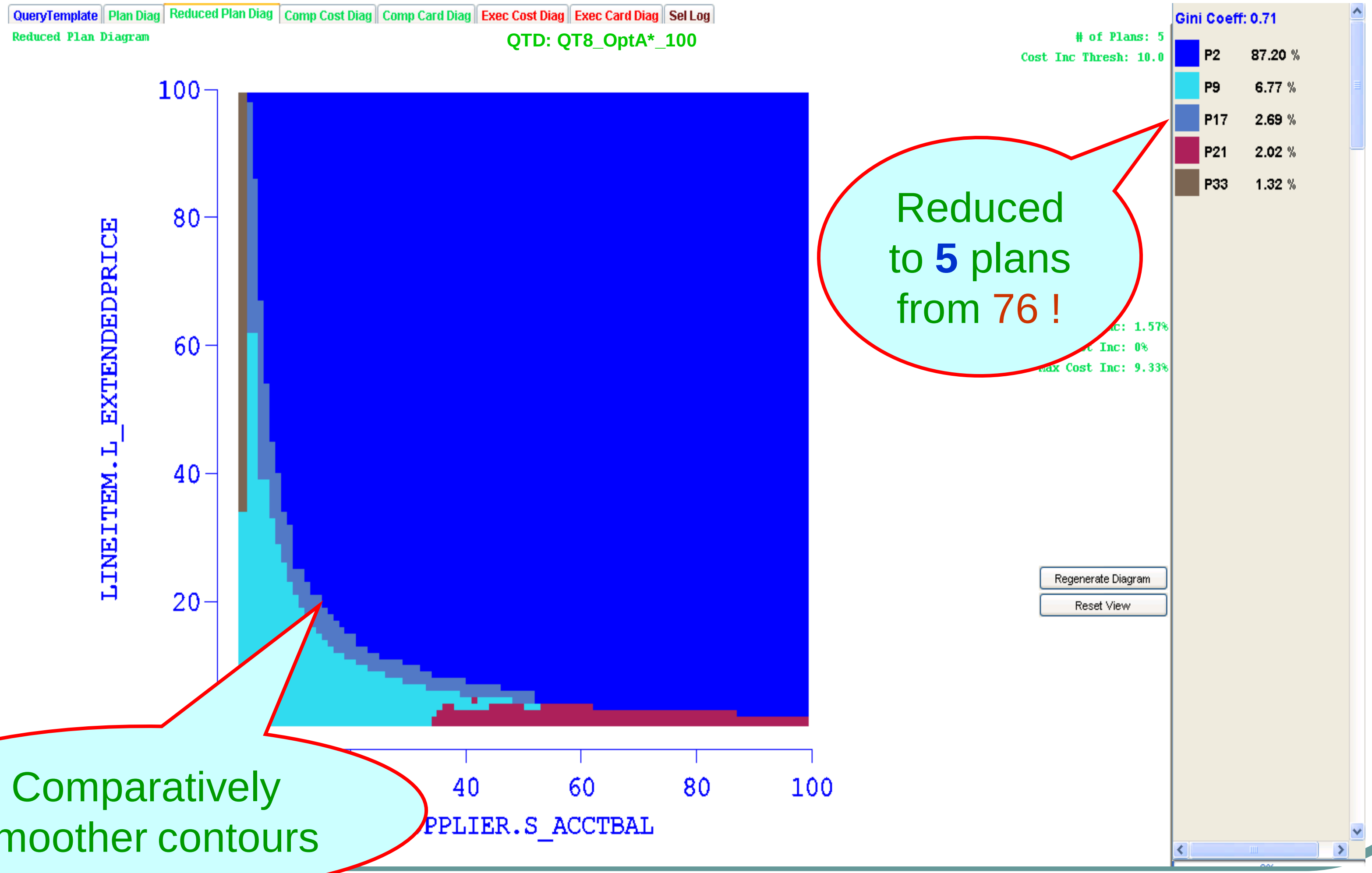
- Designed CostGreedy, a greedy heuristic-based algorithm with following properties:

  [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

# Reduced Plan Diagram [λ=10%]

[QT8, OptA*, Res=100]



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

# Anorexic Reduction

Extensive empirical evaluation with a spectrum of multi-dimensional TPCH-based 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)!

# Applications of 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 selectivity parameter space

# Part III: Identifying Robust Plans with Plan Diagram Reduction

# Selectivity Estimation Errors

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

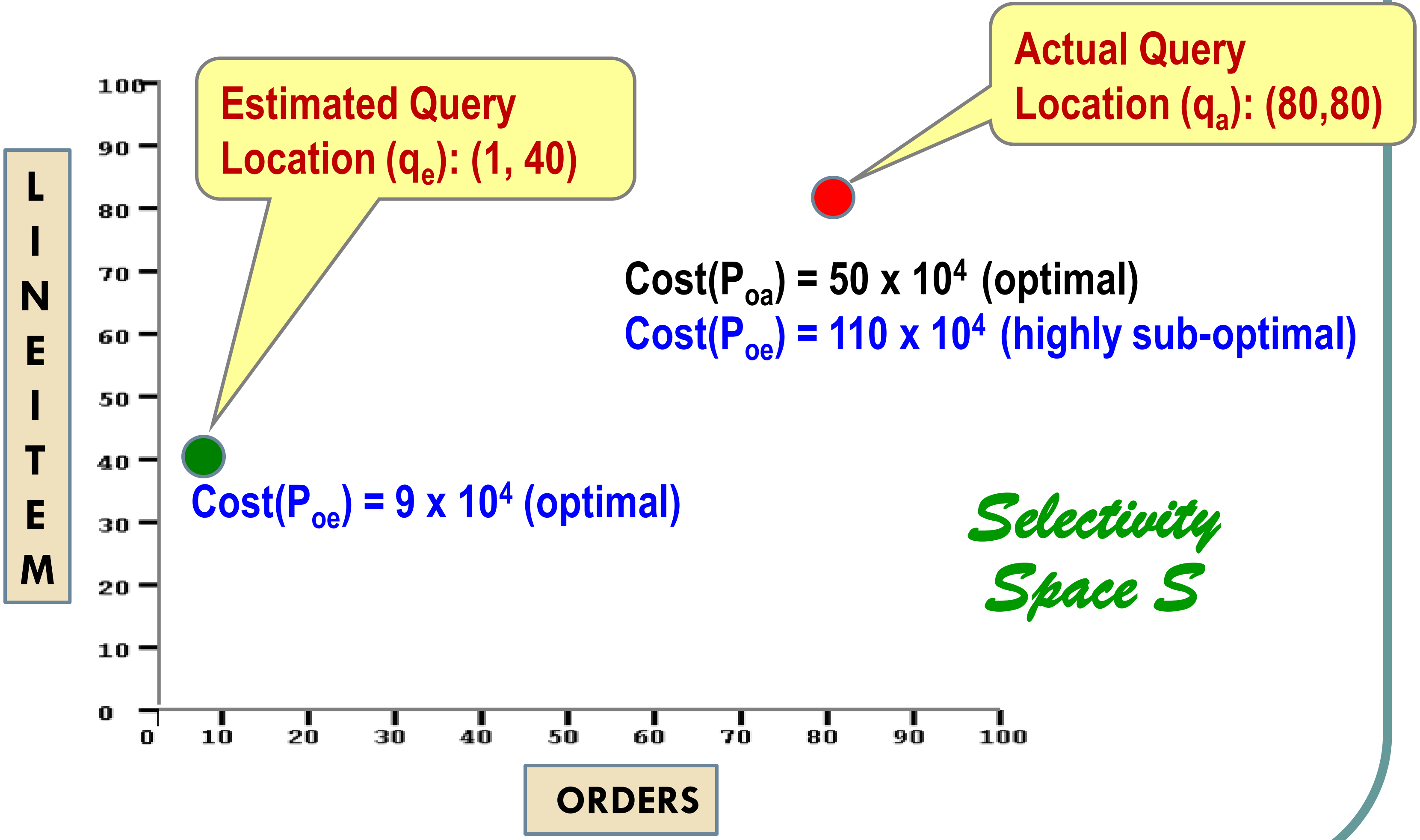
$q_a(x_a, y_a)$: actual location during execution

The difference could be substantial due to

- Outdated Statistics (expensive to maintain)
- Coarse Summaries (histograms)
- Attribute Value Independence (AVI) assumptions

- Chronic problem in database design

# Impact of Error Example



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

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

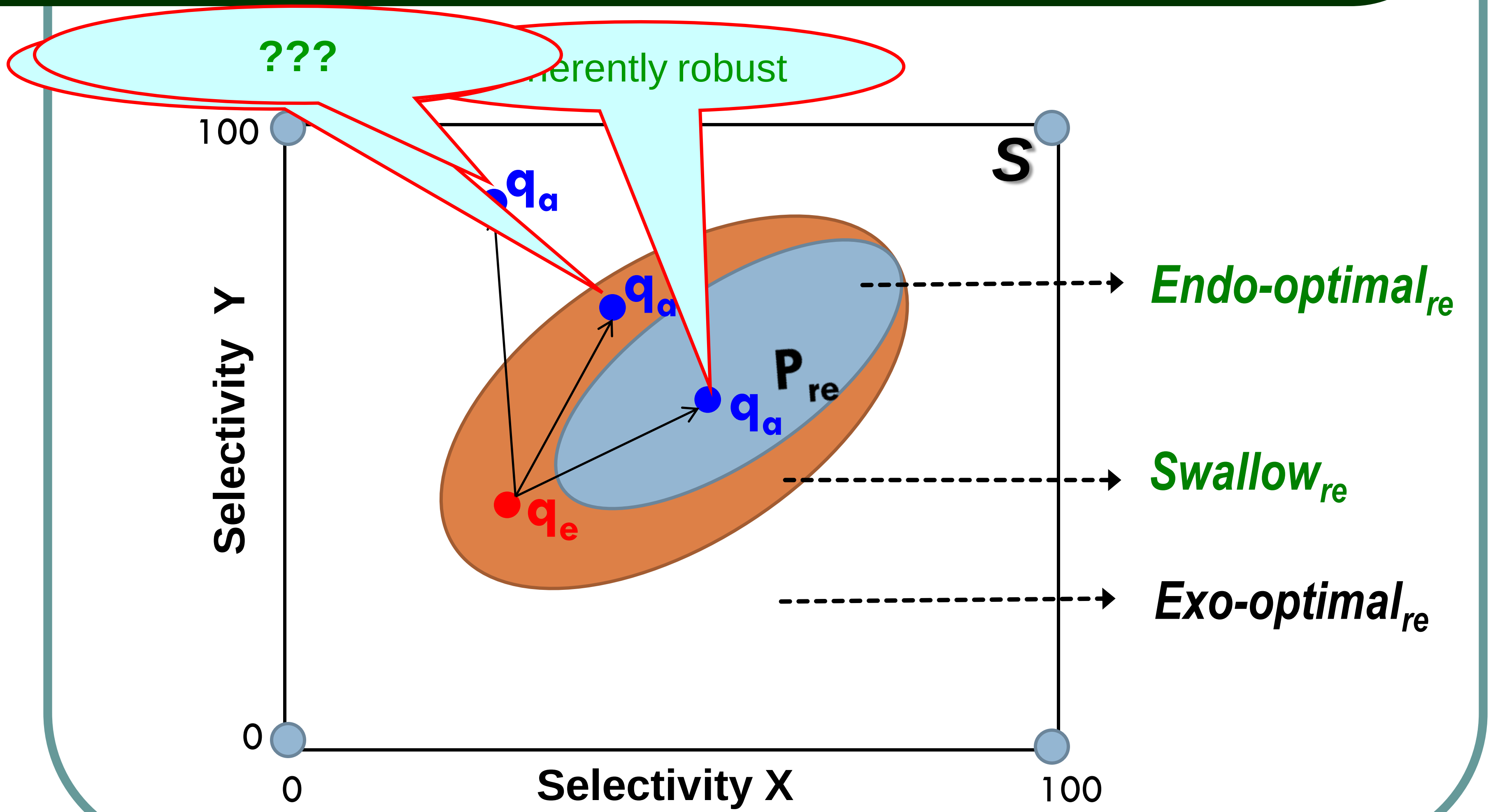
Cost($P_{oa}$) = 50 x $10^4$ (optimal)

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

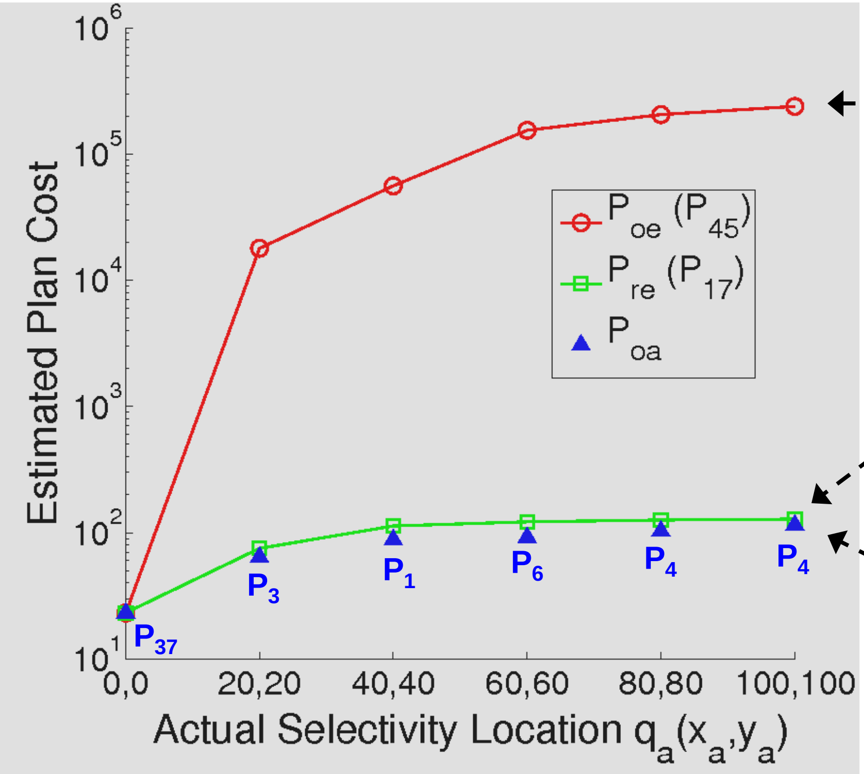


# Positive Impact of Reduction

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 ($P_{re}$)**

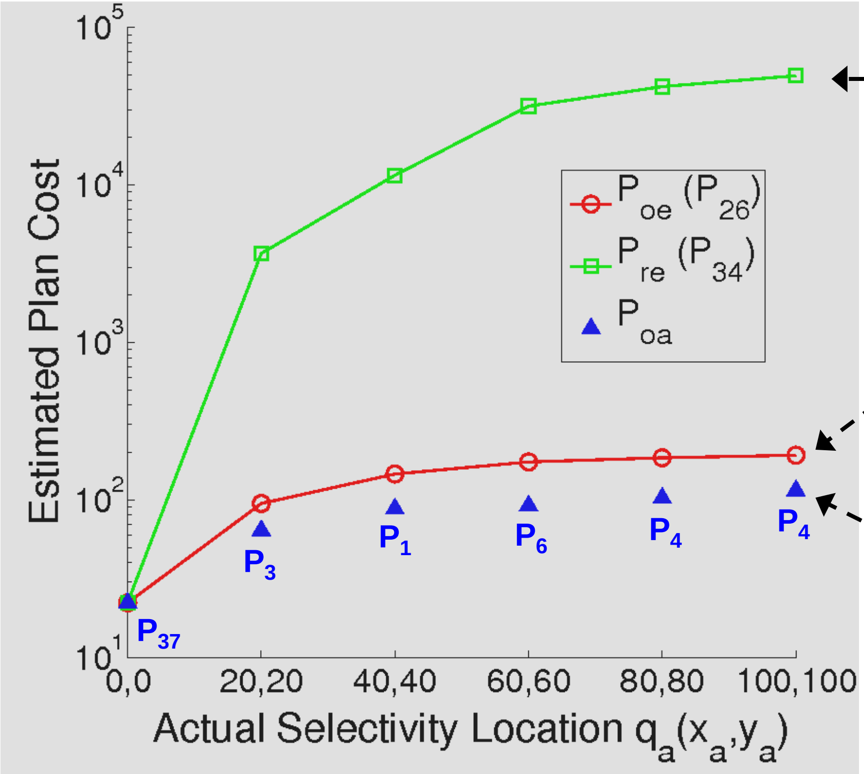
**Local Optimal Plan ($P_{oa}$)**

# 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?

# Analogy Update

India can swallow Sri Lanka only if Indian passport can guarantee cost of living of Lankan citizen is within $\lambda$ of that obtained with the Lankan passport, no matter where the Lankan citizen emigrates.

# Our Contribution

Mathematical analysis to show that only the perimeter of the selectivity space suffices to determine global safety

Border Safety ⇒ Interior Safety !

# CONCLUSIONS

# Picasso Visualizer

- Conceived and developed the **Picasso tool** for automatically generating plan, cost and card diagrams
  - optimizer debugger / research platform / teaching aid

- Analyzed representative plan diagrams on popular commercial query optimizers
  - Optimizers make *fine grained* choices
  - Plan optimality regions can have *intricate patterns* and *complex boundaries*
  - *Non-Monotonic* cost behavior exists where increasing input and result cardinalities decrease the estimated cost
  - Basic assumptions of PQO research literature on PQO *do not hold* in practice; hold approximately for reduced plan diagrams

# Take Away

We can efficiently produce plan diagrams that simultaneously possess the desirable properties of being online, anorexic, safe and robust.

This result could play a meaningful role in designing the next generation of database query optimizers.

# More Details

http://dsl.serc.iisc.ernet.in/projects/PICASSO

Publications, Software, Sample Diagrams

# QUESTIONS?

# END PRESENTATION