Parametric Query Optimization for Linear and Piecewise Linear Cost Functions

Arvind Hulgeri
S. Sudarshan
Dept of Computer Science and Engg.
Indian Institute of Technology Bombay
Parametric query: An example

Select *
From A, B
Where A.x = B.y and A.z < ? And B.w < ?

- E.g. Cost function $f = a_1.s_1 + a_2.s_2 + a_3$
  - Where, $s_1 = \text{selectivity of predicate } “A.z < ?”$
  - $s_2 = \text{selectivity of predicate } “B.w < ?”$
  - $a_1, a_2, a_3$ are constants
- (using merge-join assuming relations are sorted on join attribute)
Conventional Opt v/s PQO

- Conventional optimization
  - Assumes complete knowledge of all cost parameters
    - E.g. selectivity and resource availability
  - Generates a single optimal plan for a given query

- Parametric query optimization (PQO)
  - Generates multiple candidate plans, each optimal for some region of the parameter space
    - POSP: Parametrically optimal set of plans
  - Picks appropriate plan at run time
PQO: A 1-parameter example

\[ R(p) = \text{region of optimality of plan } p \]
Overview

- We classify cost functions as:
  - linear, piecewise linear and non-linear
- PQO for linear cost functions
  - Recursive decomposition algorithm
  - Cost polytope algorithm
- PQO for piecewise linear cost functions
  - Extend a conventional query optimizer
- Non-linear cost functions approximated by piecewise linear cost functions
PQO for Linear Cost Functions

- Our solutions use a conventional optimizer as a subroutine
- The solutions work for arbitrary number of parameters
- Assumption: The conventional optimizer returns the cost function of the optimal plan
Polytope Examples

- Convex polytope = intersection of halfspaces
Properties of Linear Cost Functions [Ganguly, VLDB98]

- If all the vertices of a polytope in the parameter space have same optimal plan then the plan is optimal at all points within that polytope.
- Each plan in POSP has only one region of optimality and the region is a convex polytope.
Recursive Decomposition Algorithm

- Start with the parameter space of interest— a convex polytope
- Optimize the vertices of the polytope using a conventional query optimizer
- If two of the vertices of a polytope have two different optimal plans then
  - Partition the polytope into two polytopes
  - Continue recursively
Shortcomings of the recursive decomposition algorithm

- May overpartition the parameter space and may need to merge partitions in a postpass.
- We can reduce number of calls to the conventional optimizer using cost polytope algorithm
Cost Polytope Algorithm

- Based on an online polytope construction algorithm
- The cost function of each plan is represented by a hyperplane in $\mathbb{R}^{n+1}$
  - N parameter dimensions + 1 cost dimension
- Construct the cost polytope
  - A lower convex polytope that represents the optimal cost at each point in the parameter space
Cost Polytope: An Example

Parameter

$\text{R}(p) = \text{region of optimality of plan } p$
Cost Polytope Algorithm

- Start with a initial cost polytope
- Put vertices of the parameter space polytope into a queue of vertices to be optimized
- Repeat till the queue is empty
  - Remove and optimize the first vertex in the queue
  - Intersect the cost hyperplane with the cost polytope
  - Project new vertices of the cost polytope onto parameter space and insert the projection points into the queue
Cost polytope algorithm: An example

Parameter

Cost

- Not optimized
- Currently optimized
- Already optimized
Cost polytope algorithm: An example

Cost

Parameter

- Not optimized
- Currently optimized
- Already optimized
Cost polytope algorithm: An example

Cost

Parameter

- Not optimized
- Currently optimized
- Already optimized
Cost polytope algorithm: An example

Parameter

Cost

Not optimized
Currently optimized
Already optimized
Cost polytope algorithm: An example

![Cost polytope diagram](image)

- **Not optimized**
- **Currently optimized**
- **Already optimized**

Parameter

Cost
Cost polytope algorithm: An example

Cost

Parameter

- Not optimized
- Currently optimized
- Already optimized
Faces and facets of a polytope

3-D polytope
facets = 2-faces

N-D polytope
facets = (N-1)-faces

faces = \( \bigcup_{i=0}^{N-1} i\text{-faces} \)

F = |faces|
f = |facets| = |POSP|
v = |0-faces|
Complexity of Cost Polytope Algorithm

- Cost polytope algorithm makes a maximum of $F$ calls to the optimizer.
- The lower bound on the number of calls is $v$.
- Under certain assumptions, the expected number of calls is $(f + v)$.
- In general, in high-dimension, $f << v$. 
PQO solutions for linear case do not extend to piecewise linear case
Piecewise Linear Cost Function

- Partition the parameter space into convex polytopes
  - Within each partition the cost function is linear in the parameters
- But pre-partitioning the space to make all cost functions linear in each partition is impractical
Extend a conventional query optimizer (System-R or Volcano)
- Extensions are intrusive to the query optimizer
- Partition space only when necessary ("on demand")
- Extend plan cost:
  - Cost $\rightarrow$ Cost function
- Extend comparison of alternative operators or plans
  - Pick min cost plan $\rightarrow$ MinMergeCostFunctions

Extensions work for arbitrary number of parameters
MinMergeCostFunction: An example

Cost

Parameter
MinMergeCostFunction: An example

![Graph showing the cost and parameter relationship for MinMergeCostFunction. The graph illustrates how the cost decreases as the parameter increases, with some specific points indicated on the graph.](image-url)
Extending System-R Algorithm

- Extended System-R algorithm is exactly same as basic System-R algorithm except:
  - Replace cost by cost function
  - Use AddCostFunction instead of simple cost addition
  - Use MinMergeCostFunction instead of simple cost comparison
Related Work

- Graefe and Karen [SIGMOD'89], Cole and Graefe [SIGMOD'94], Ioannidis, Ng, Shim and Sellis [VLDB'92]
- Ganguly and Krishnamurthy [COMAD'94]
- Sumit Ganguly [VLDB'98]
- Sumit Ganguly, A Framework for Parametric Query Optimization, Unpublished manuscript; Personal Communication, 2001
Conclusion

PQO for linear cost functions:
- Simple and minimally intrusive
- Works for arbitrary number of parameters

PQO for piecewise linear cost functions
- Intrusive
- Works for arbitrary number of parameters
- Very general since nonlinear and discontinuous cost functions can be approximated to piecewise linear form