A genetic algorithm for Database Query Optimization

Kristin Bennett, Michael C. Ferris Yannis E. Ioannidis

1991
The problem

• Query optimization is getting slower and slower…
  – More relations are being joined
  – Current techniques are inadequate to support some of the emerging database applications
Query optimization

- QUERY q
- Set of all strategies appropriate to answer the query
- Strategy s where c(s) = min
- Optimized query q
- Database

Optimizer
Strategy/Execution spaces

• What are they?

• Search space
Processing trees of interest

Left-deep tree
(Ł space)

Bushy tree
(A space)
The System-R algorithm

• Strategy for creating the optimal solution
  …

• Only left-deep trees are considered

• Too slow when many relations need to be joined
Genetic Algorithms

- A search technique used to find exact or approximate solutions to optimization and search problems

- Steps:
  - Initialization
  - Selection
  - Reproduction
  - Termination
Local neighbourhood algorithm

repeat
    for each chromosome \( i \) do
        evaluate \( f(i) \)
        broadcast \( f(i) \) in the neighbourhood of \( i \) /*3 hops away*/
        receive \( f(j) \) for all chromosomes \( j \) in the neighbourhood
        select chromosome \( k \) to mate from the neighbourhood of \( i \) based on fitness
        reproduce using chromosomes \( i \) and \( k \)
        replace chromosome \( i \) with one of the offspring /*if they are better than it*/
    until population variance is small
Left-deep strategies $\mathcal{L}$

- Frequently a strategy exists in $\mathcal{L}$ which is very close to the optimal
  - The search space is much smaller
  - However it cannot beat System-R

- Each chromosome represents a left-deep strategy

- Mutation
  - Randomly change the join method
  - Swap two adjacent genes
Bushy strategies

- Often the best strategies are here but it’s too costly to look for them

\[ k^{J_O} \]

- \( k \) – join number; \( J \) – join method; \( O \) – orientation

- No Cartesian products
  The representation is based on labeling joins and not relations

- Mutation
  - Change the join method or orientation
  - Swapping of neighbouring genes

\[ 1^{J_A} \ 5^{J_R} \ 2^{J_R} \ 3^{J_A} \ 4^{J_A} \]
Crossover

Modified Two Swap

\[ X = m_A^n C^m B^m D^m F^n E\]
\[ Y = m_B^n C^m D^m F^m E^n A\]
\[ X_{\text{new}} = m_D^n C^m B^n A^n F^n E\]

CHUNK

\[ X = 1_A^n 5_R^n 2_A^m 3_A^m 4_A^n\]
\[ Y = 3_R^m 5_A^n 1_R^m 4_A^m 2_A^m\]
\[ X_{\text{new}} = 5_A^n 1_R^m 2_R^m 3_R^m 4_A^m\]
Results
Output quality

- GA in L with CHUNK
- GA in L with M2S
- System-R in L
- GA in A with M2S
- GA in A with CHUNK

Graphs showing the relationship between the number of relations and the scaled cost for different algorithms and configurations.
Time

![Graph showing time vs number of relations for different algorithms (GA in $L$ with CHUNK, GA in $L$ with M2S, System-R in $L$, GA in $A$ with M2S, GA in $A$ with CHUNK).]
Questions?