Query Planning for Searching Inter-Dependent Deep-Web Databases

Fan Wang[1], Gagan Agrawal[1],
Ruoming Jin[2]

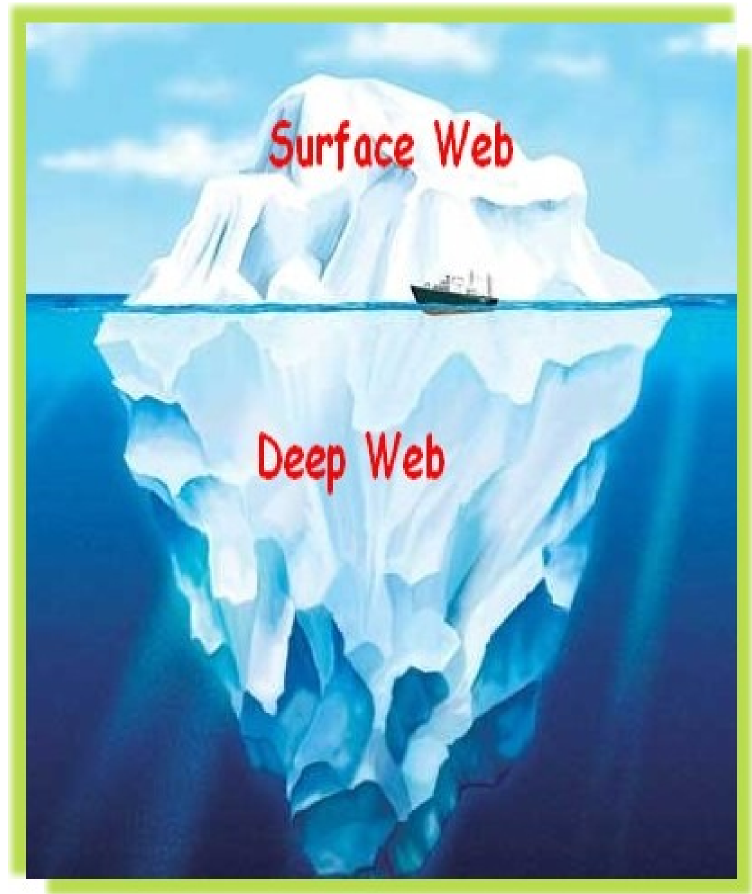
[1] Department of Computer Science and Engineering
Ohio State University, Columbus, OH, USA 43210

[2] Department of Computer Science
Kent State University, Kent, OH, USA 44242

wangfa@cse.ohio-state.edu

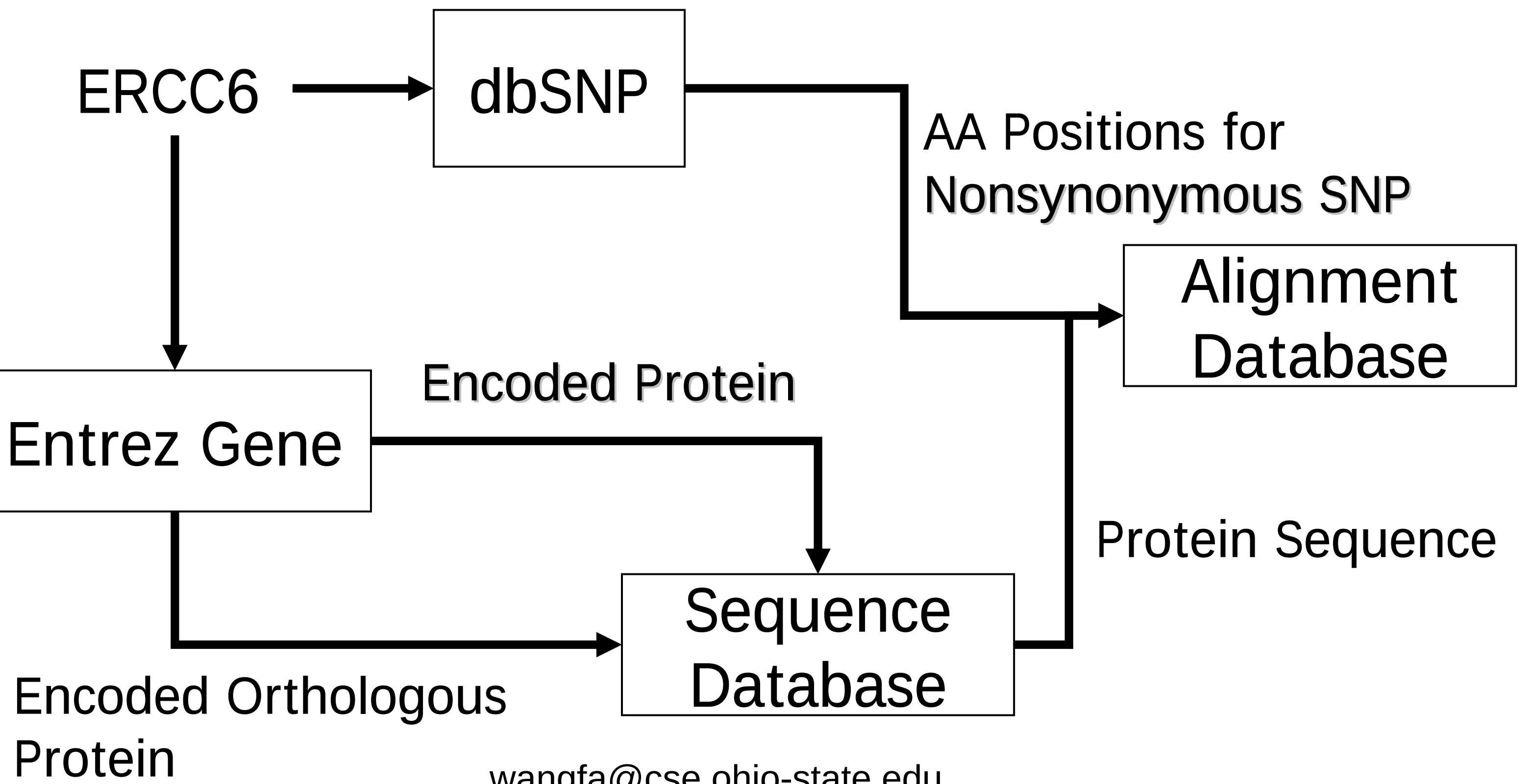
Introduction

- The emergence of deep web
  - Deep web is huge
  - Different from surface web
  - Challenges for integration
    - Not accessible through search engines
    - Inter-dependences among deep web sources



Motivating Example

Given a gene **ERCC6**, we want to know the **amino acid occurring in the corresponding position in orthologous gene of non-human mammals**



Observations

- Inter-dependences between sources
- Time consuming if done manually
- Intelligent order of querying
- Implicit *sub-goals* in user query

Contributions

- Formulate the query planning problem for deep web databases with dependences
- Propose a dynamic query planner
- Develop cost models and an approximate planning algorithm
- Integrate the algorithm within a deep web mining tool: SNPMiner

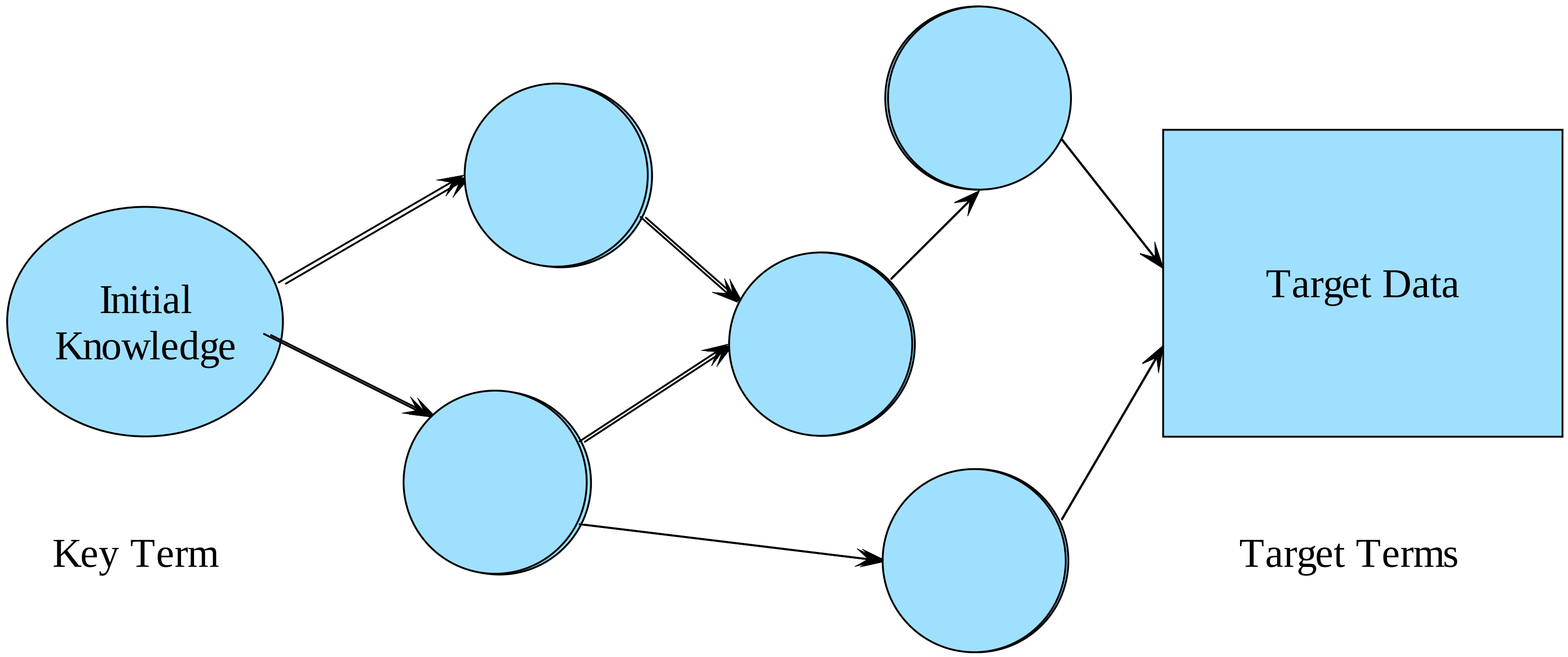
Roadmap

- Introduction and Motivation
- Problem Formulation
- Planning Algorithm
- Evaluation
- Related Work
- Conclusion

Formulation

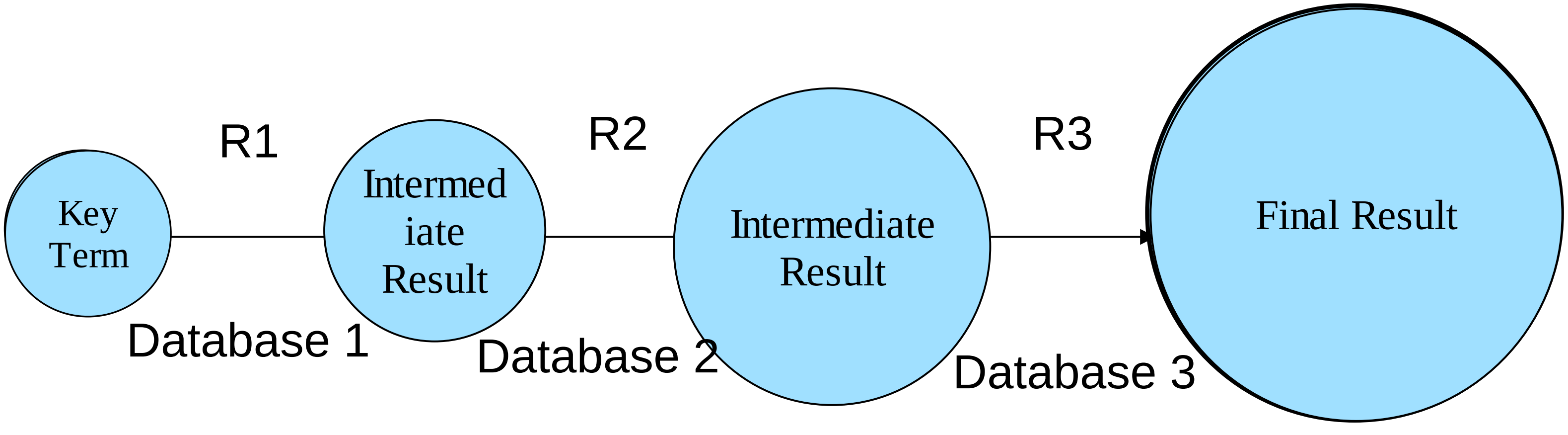
- Universal Term Set $T = \{t_1, t_2, ..., t_n\}$
- Query Q is composed of two parts
  - Query Key Term: focus of the query (ERCC6)
  - Query Target Terms: attributes of interest (Alignment)
- Data sources
  - Each data source D covers an output set
  - Each data source D requires an input set
- Find a query plan, an ordered list of data sources
  - Covers the query target terms with maximal ***benefit***
  - As short as possible
  - NP-Complete problem

Problem Scenario



Production System

- Working Memory
- Target Space
- Production Rules
- Recognize-Act Control



wangfa@cse.ohio-state.edu

Roadmap

- Introduction and Motivation
- Problem Formulation
- Planning Algorithm
- Evaluation
- Related Work
- Conclusion

Algorithm

- Dependency Graph
- Planning Algorithm Detail
- Benefit Model

Dependency Graph

- Dependency relation $\prec_{DR}$
  - Format: $\{D_i, D_{i+1}, \ldots, D_{i+m}\} \prec_{DR} D_j$
  - Hypergraph
    - Hyperarc: ordered pair (parents, child)
    - AND node
    - Neighbors

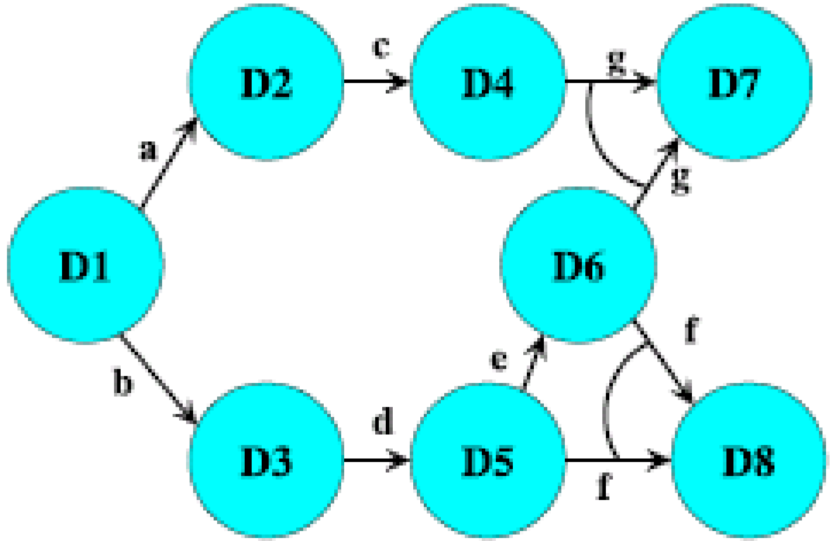


Fig. 1. Dependency Graph Example

Concepts

- Database Necessity (DN)
  - Each term is associated with a DN value
  - Measures the extraction priority of a term and the importance of a database scheme
  - For term t, if k database schemes can provide it, the DN value is $\frac{1}{k}$

Concepts

- Hidden Nodes
  - Nodes connecting current working state and the target space
- Partially Visualize Hidden Nodes
  - Multiple layers of hidden nodes bring difficulty

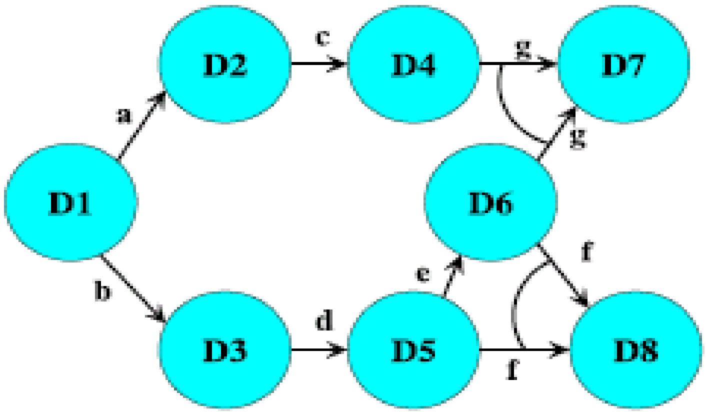


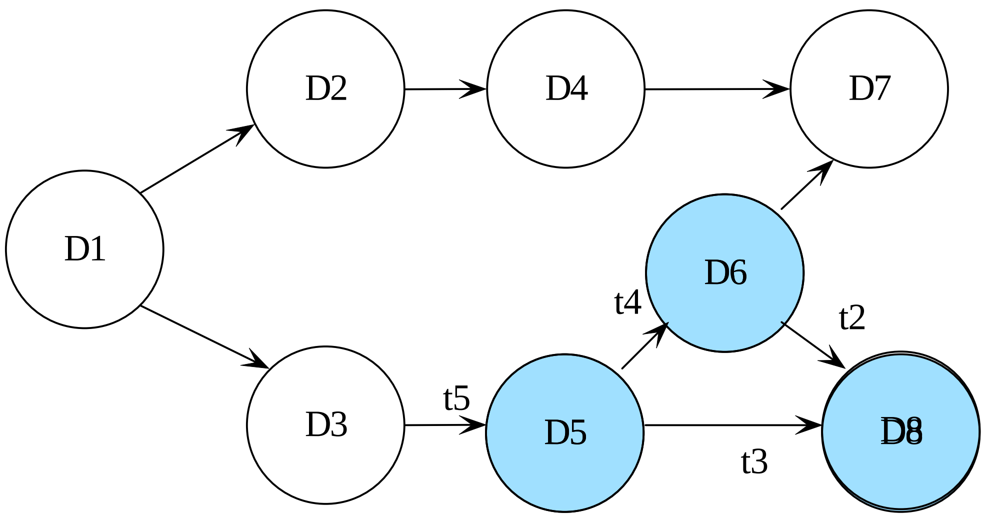
Fig. 1. Dependency Graph Example

wangfa@cse.ohio-state.edu

Visualize Hidden Nodes

- Target Space Enlargement

Target Space: {{t1,t}2,t3}t4}t5}



1. Find a target term t with DN=1
2. Visualize the database D which provides t
3. Add D's input set to target space
4. Repeat above steps untill done

Planning Algorithm Detail

Find_TopK_Query_Plans(PR, WS, TS)

while enlargeable(WS)
  Enlarge WS
Initialize queue Q and P

while $size(Q) \leq K$

  if $(\exists e \in TS$ **and** $e \notin WS)$
  and $(\exists r \in PR$ **and** $\exists o \in O(r)$ **and** $o \in TS)$

    Find candidate rule set CR

    foreach $r \in CR$
      Compute benefit score according to benefit model
    Select r_opt, the rule with the highest benefit

    Add r_opt to P, and update WS

  else Add P to Q and re-order Q
return (Q)

Planning Algorithm Detail

- The approximation ratio of our greedy algorithm is $\frac{|R|+1}{2|R|}$

Benefit Model

- Select an appropriate rule at each iteration of the planning algorithm
- Four metrics
  - Database Availability
  - Data Coverage (DC)
  - User Preference (UP)
  - Potential Importance (PI)

Model Metrics

- Data Coverage
  - Number of target terms covered by the current rule exclusively
- User Preference
  - Domain users have preference for certain database (rule) for a particular term
  - Weighted sum of user preference values of unfound target terms

Model Metrics

- Potential Importance
  - Some databases are more important due to their linkings to other important databases (e.g.)
  - The more number of “important” databases can be reached from the current database, the higher the potential importance of the current database is

Roadmap

- Introduction and Motivation
- Problem Formulation
- Planning Algorithm
- Evaluation
- Related Work
- Conclusion

Experimental Setup

- SNPMiner System
  - Integrates 8 deep web databases
  - Provides a unified user interface
- Experimental Queries

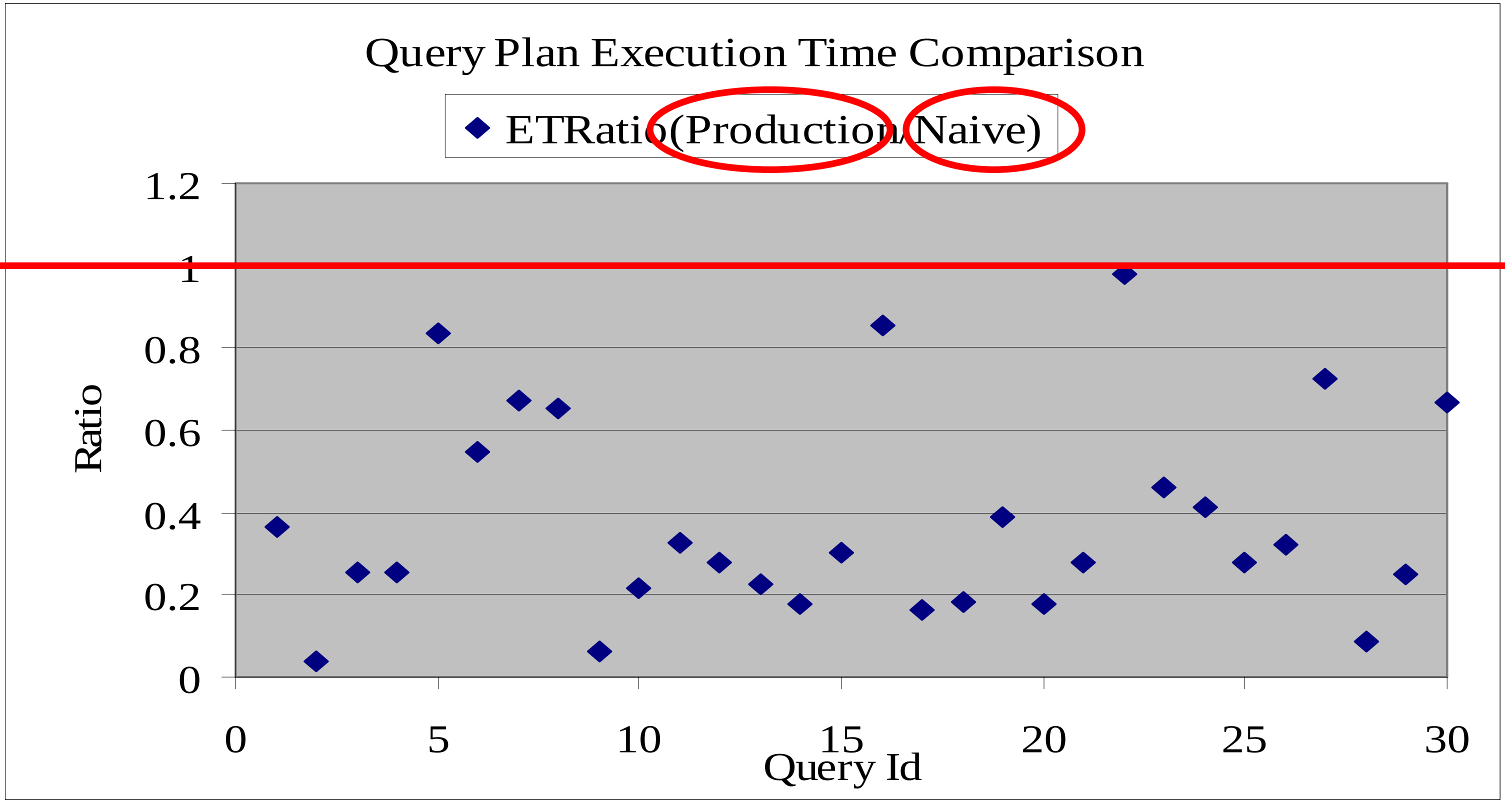
Table 2. Experimental Query Statistics

Query ID	Number of Terms
1-8	2-5
9-16	8-12
17-24	17-23
25-28	27-33
29,30	37-43

Planning Algorithm Comparison

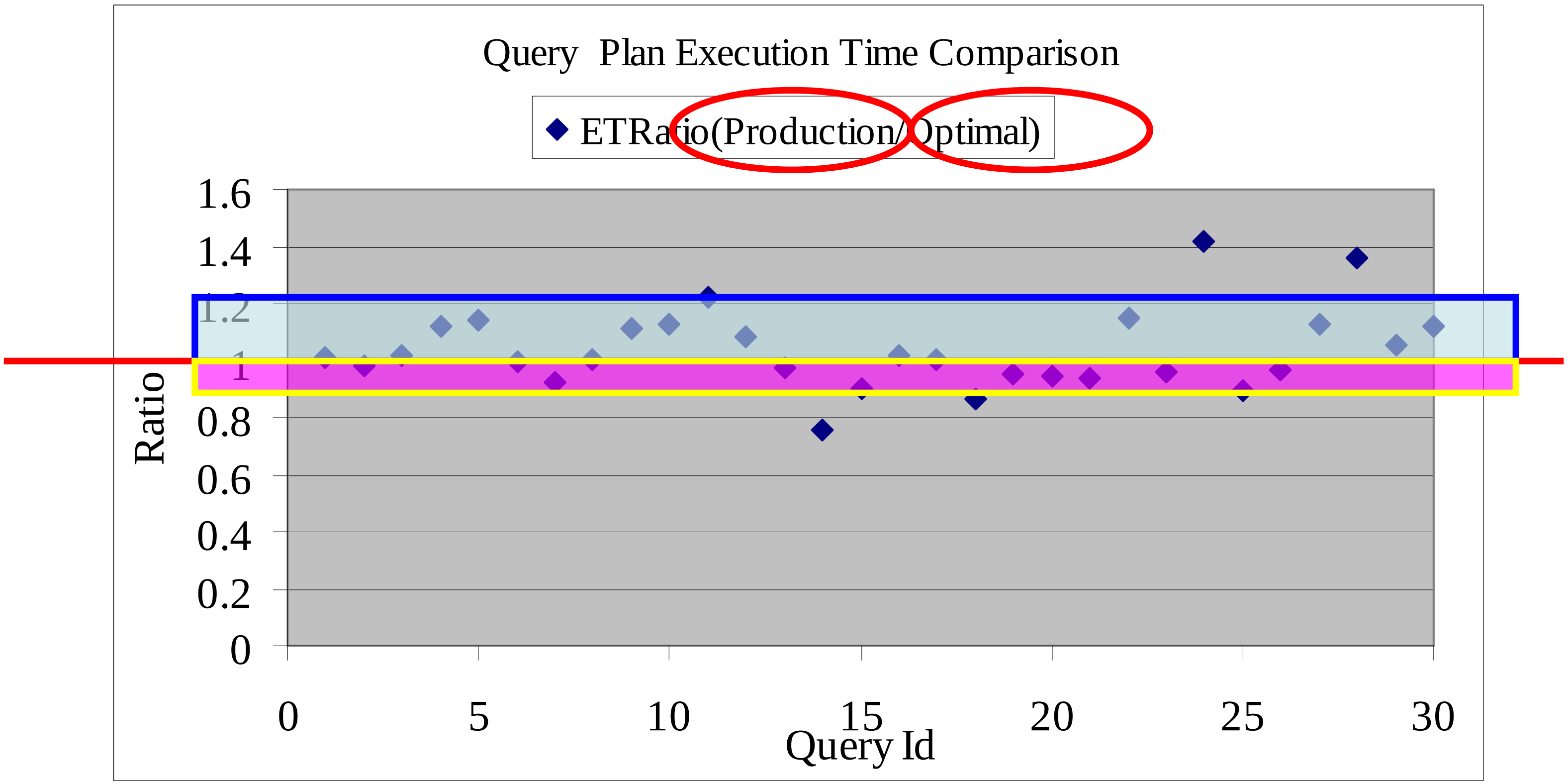
- Naïve Algorithm (NA)
  - Select all rules which can be fired at each iteration until all requested terms are covered
  - No rule selection strategy used
- Optimal Algorithm (OA)
  - Search the entire space (exhaustive algorithm)
  - The plan with the least execution time
- Production Rule Algorithm (PRA)

PRA vs. NA



1. All ratio data points smaller than 1
2. PRA generates much faster query plans than NA
3. Queries with shorter plans benefit more from PRA

PRA vs. OA



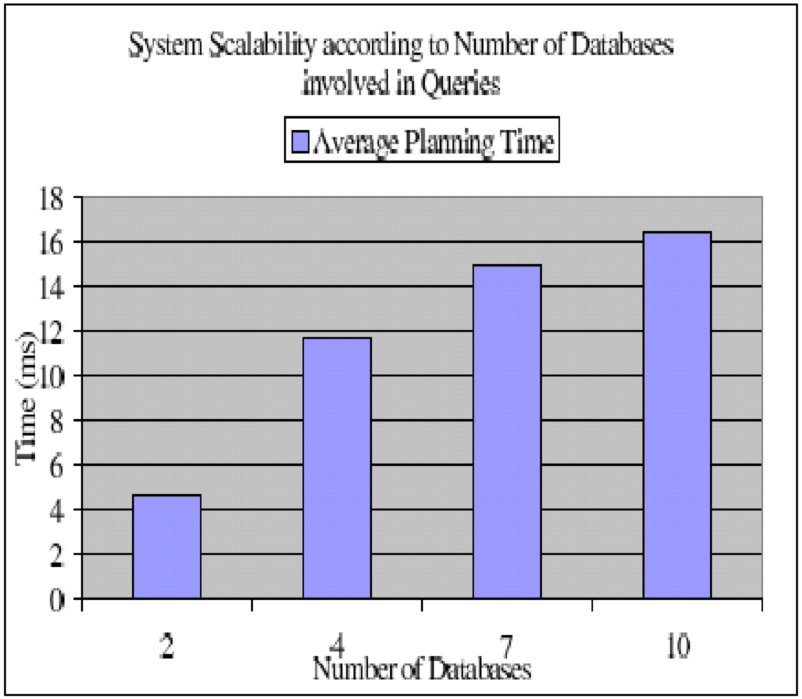
1. All ratio data points distributed around 1
2. In terms of query plan execution time, PRA has performance close to OA
3. For most cases, the execution of the plans from PRA is no more than 20% slower than the plans from OA
4. For some cases, the plans from PRA run faster than the plans from OA (no more than 10% faster) due to the variation of database response time
5. In most cases, PRA generates exactly the same plan as OA

wangfa@cse.ohio-state.edu

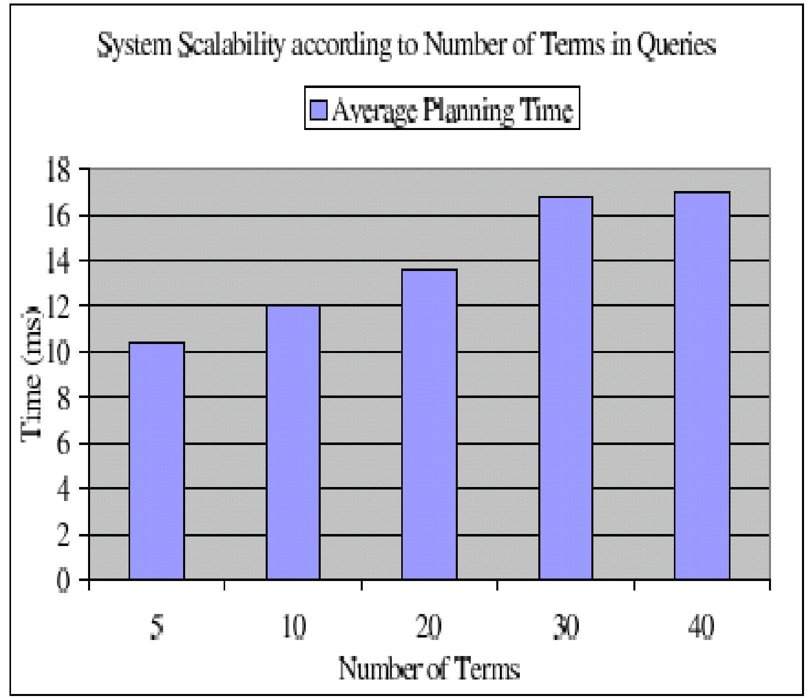
Enlarge Target Space

1. Query plans generated with enlargement run faster
2. Query plans generated with enlargement are shorter

Scalability



(a)



(b)

Our system scales well to the number of databases and the number of query terms

Roadmap

- Introduction and Motivation
- Problem Formulation
- Planning Algorithm
- Evaluation
- Related Work
- Conclusion

Related Work

- Query Planning
  - Navigational based query planning
  - SQL based query planning
  - Bucket Algorithm
- Deep Web Mining
  - Database selection
  - E-commerce oriented, no dependency
- Keyword Search on Relational Databases
- Select-Project-Join Query Optimization

Conclusions

- Formulate and solve the query planning problem for deep web databases with dependencies
- Develop a dynamic planning algorithm with an approximation ratio of ½
- Our benefit model is effective
- Our algorithm outperforms the naïve algorithm, and obtains optimal results for most cases

wangfa@cse.ohio-state.edu

Questions/Comments?

wangfa@cse.ohio-state.edu

Data Coverage

- The number of query target terms covered by the current rule, but has not yet been covered by previously selected rules

$$\frac{P(R_k, \neg R_1, \neg R_2, \ldots, \neg R_{k-1}, TS)}{P(TS)}$$

User Preference

- Domain users have preference for certain database (rule) for a particular term
- A collaborating biologist provides the preference values
- Term t provided by r databases $D_1, D_2, \ldots, D_r$

$$0 \le UP_t^i \le 1, \sum_{i=1}^{r} UP_t^i = 1$$

- Rule R covers the following unfound target terms $UF_1, UF_2, \ldots, UF_k$
  - Preference for R is $\sum_{i=1}^{k} DN_i * UP_i$

Potential Importance

- Some database is more important due to its linking to other important databases (e.g.)
- A database D is more important
  - Find the necessary databases which provide unfound target terms $UF_1, UF_2, \ldots, UF_k$
  - More such necessary databases can be reached from D
- The potential importance for a rule

$$\sum_{i=1}^{k} \frac{r_i}{m * DN_i}$$