UNIVERSITY
OF TRENTO - Italy
Department of Information
Engineering and Computer Science

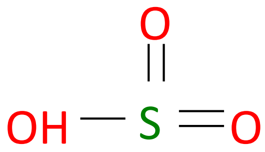
Graph Query Reformulation with Diversity

Davide Mottin, University of Trento

Francesco Bonchi, Yahoo Labs - Francesco Gullo, Yahoo Labs

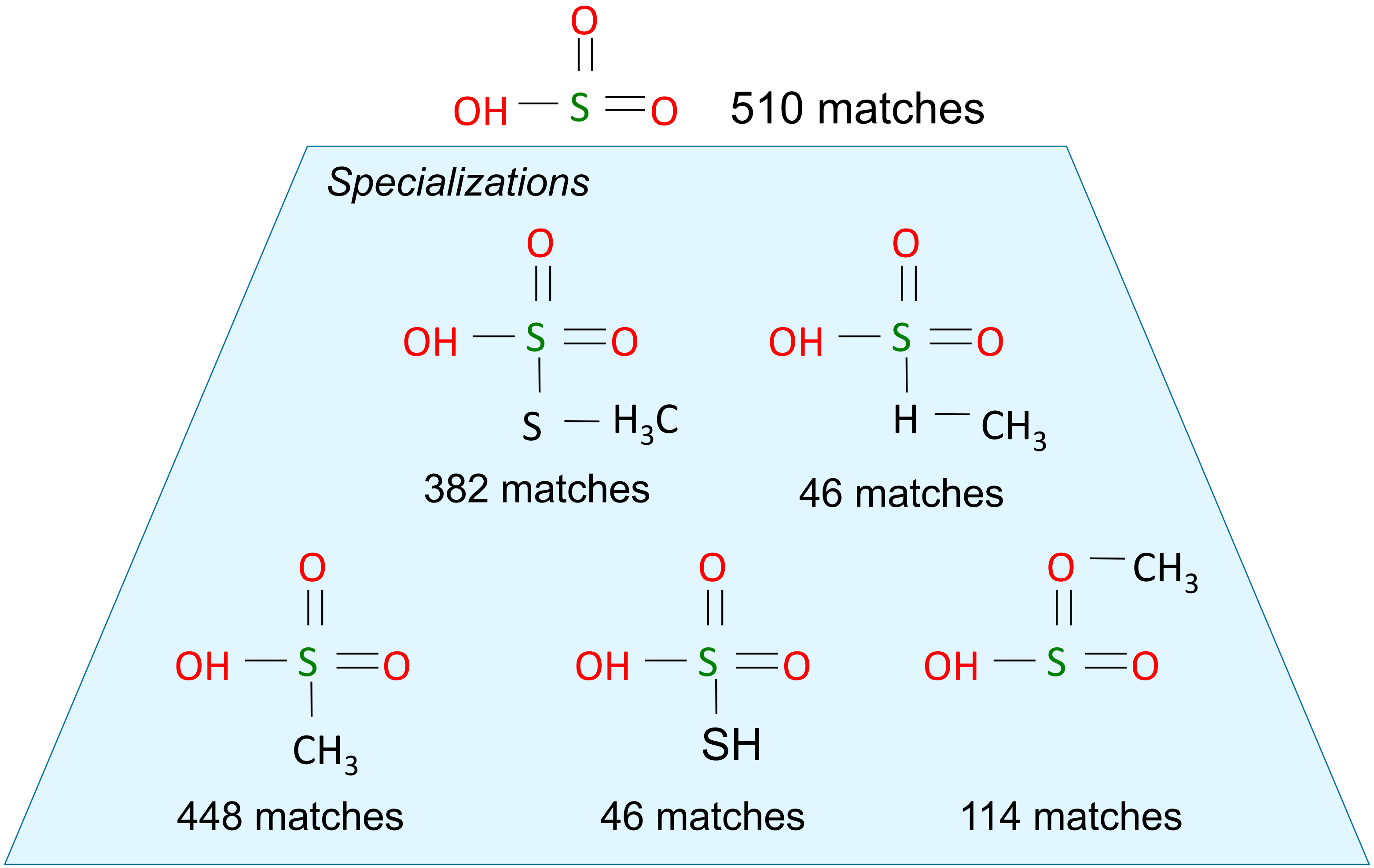
Pattern search

Query



- Too many matches
- Results are not grouped

Finding specializations



Applications

- Finding groups of molecules having a particular reagent
- Analyze a set of proteins to find diseases
- Workflow optimization
- Anomalies detection in a network
- Finding similar 3D shape search

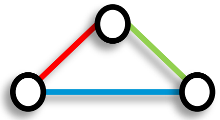
Dealing with specializations in web and relational data

- Faceted Search
  - present aspects of the results [Roy08]
- Query reformulation
  - Modify some of the query conditions
    - In structured databases [Mishra09]
    - In web search [Dang10]

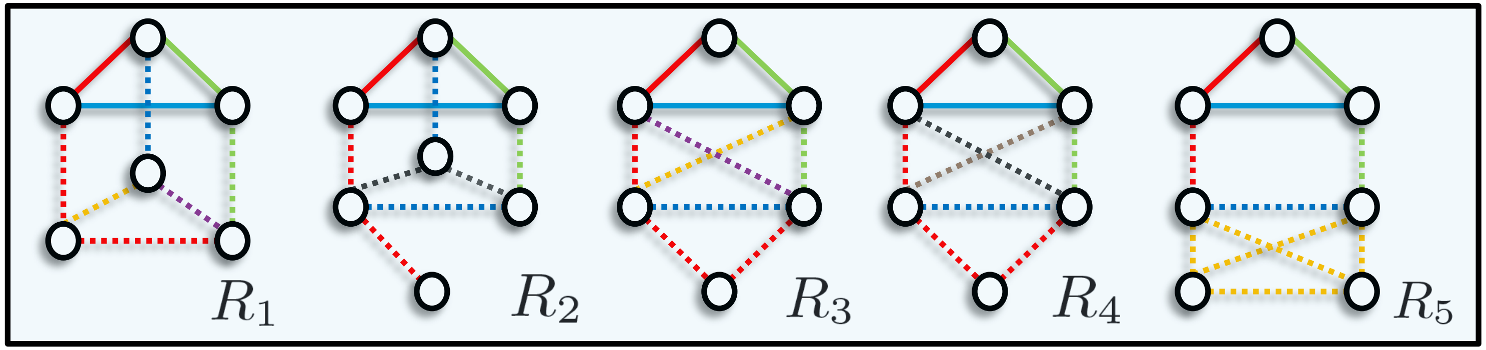
Frist Study of Problem on GRAPHS

Graph Query Reformulation

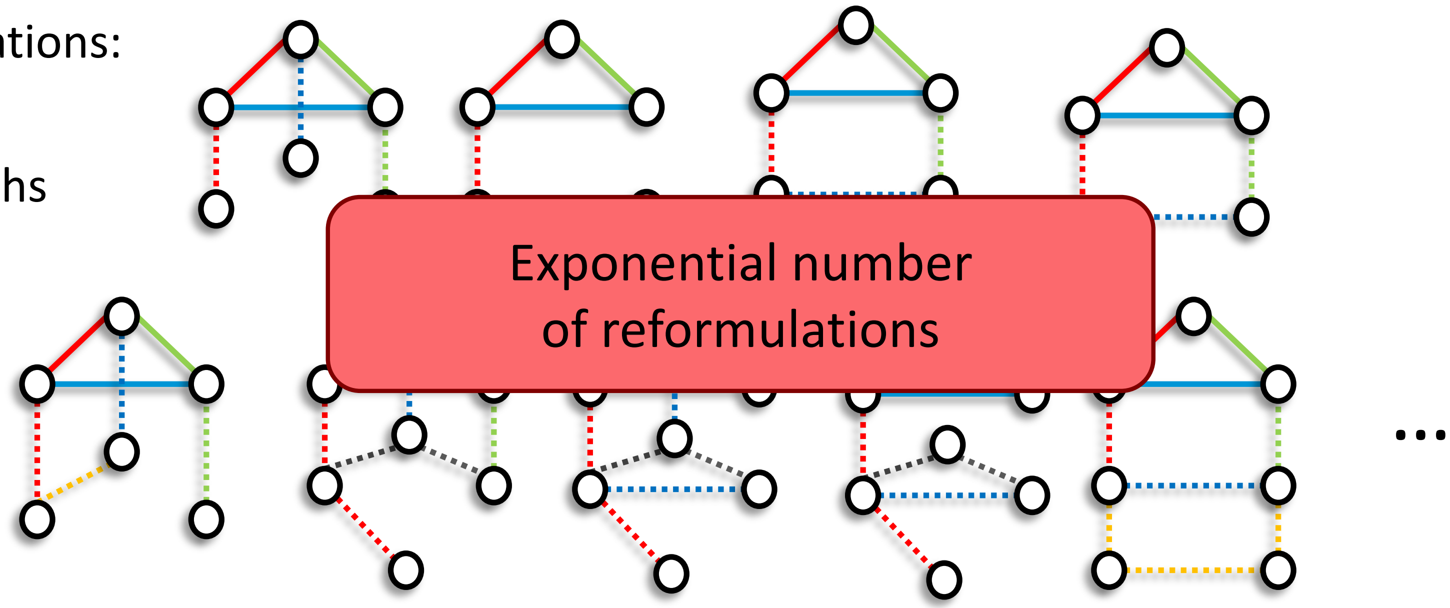
Query



Results



Reformulations:
query
supergraphs



Challenges

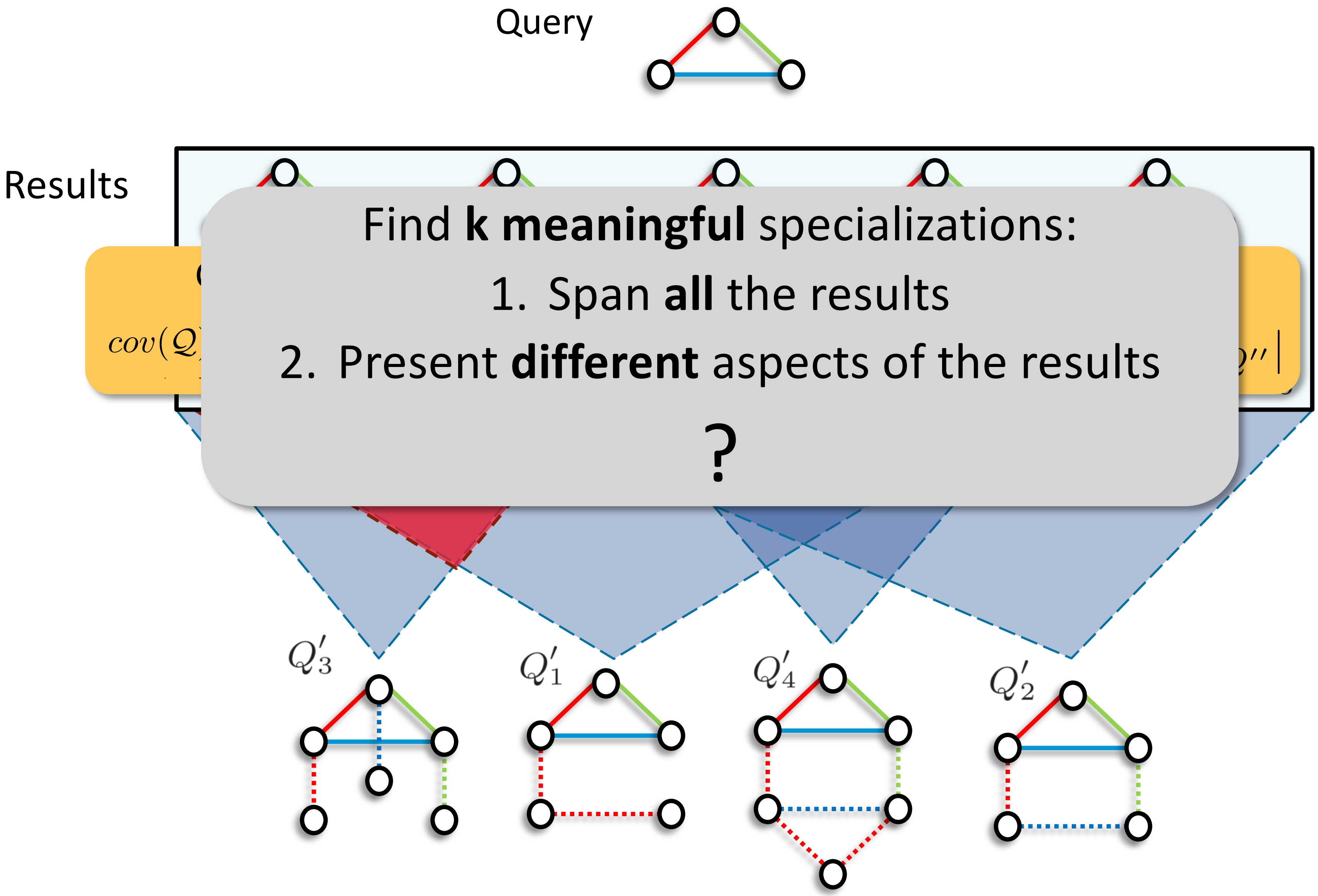
- The number of reformulation is exponential
- Quantify the interestingness of a reformulation
- Finding query reformulations is **NP**-complete

Our Approach

Graph Query Reformulation with Diversity

- Finds *k meaningful* specializations efficiently

Finding Meaningful Specializations



Graph Query Reformulation with Diversity

Problem

Find a set $\mathcal{Q}^*$ of k specializations of Q that maximize a combination of **coverage** and **diversity**

$$f(\mathcal{Q}) = cov(\mathcal{Q}) + \lambda \sum_{Q', Q'' \in \mathcal{Q}} div(Q', Q'')$$

$$\mathcal{Q}^* = \arg\max_{\mathcal{Q} \subseteq \mathbb{S}_Q} \quad f(\mathcal{Q})$$

$$\text{subject to} \qquad |\mathcal{Q}| = k.$$

Theorem (NP-hardness)
The problem reduces to **MAX-SUM Diversification** Problem, so it is NP-hard

db Trento

Solution: Greedy Algorithm

Greedy

While k-specializations are not found

1. Find the specialization leading to the maximum increment of the objective function (marginal gain)
2. Add the specialization to the results

Theorem
The algorithm is a ½-approximation

Finding the maximum gain is #P-complete [Valiant79]

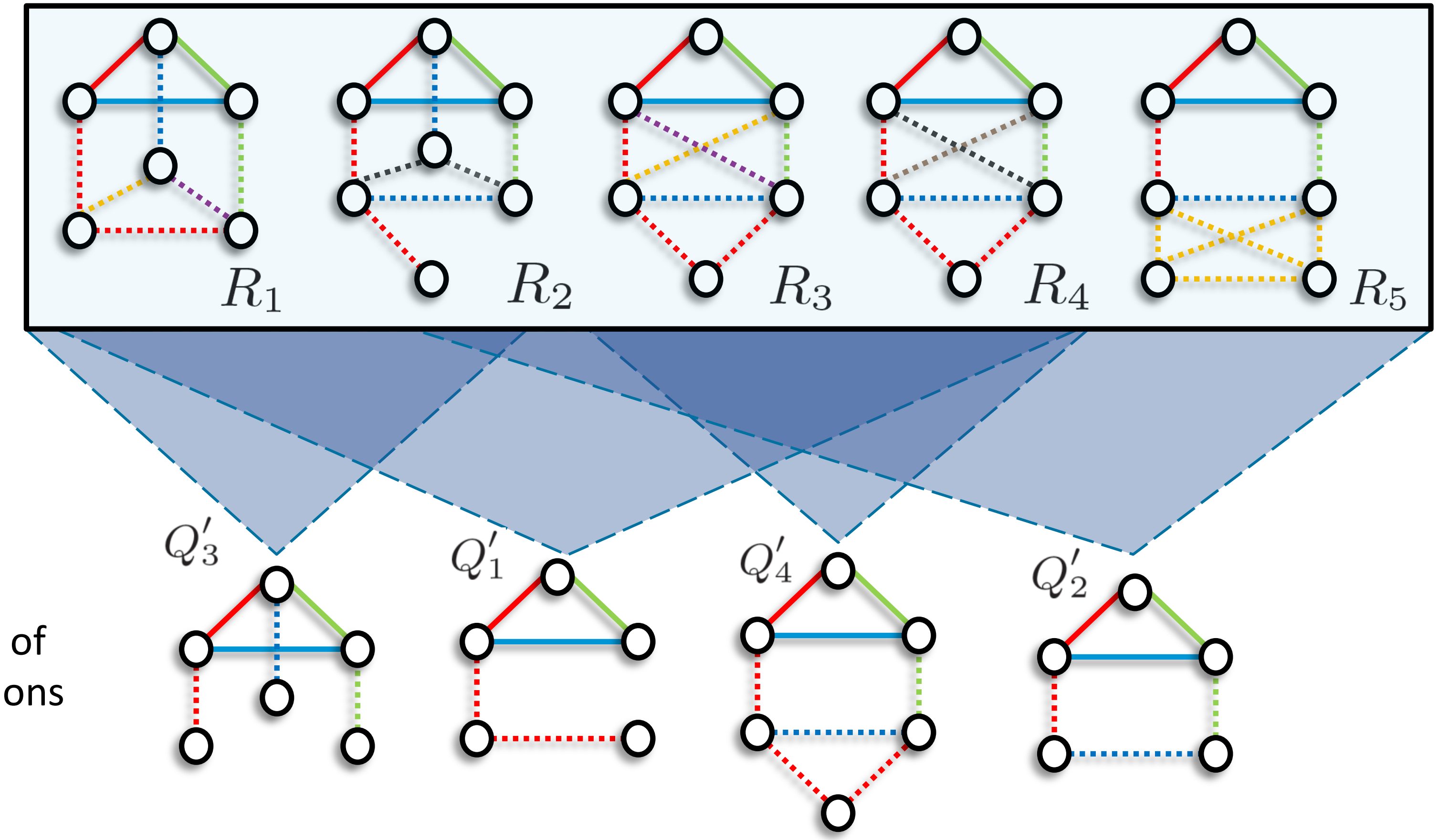
Solution

Fast_MMPG: Branch and bound algorithm with quality guarantees

The multiplicity vector

2	3	3	3	1

Results



Output set of specializations

Upper bound on the Marginal gain

Lemma

The marginal gain increases if the multiplicity of the considered item is $\leq \frac{|\mathcal{Q}|}{2}$ where $|\mathcal{Q}|$ is the number of reformulations in the reformulated set constructed so far.

Upper bound : is the value of the objective function considering only results with multiplicity $\leq \frac{|\mathcal{Q}|}{2}$

Theorem

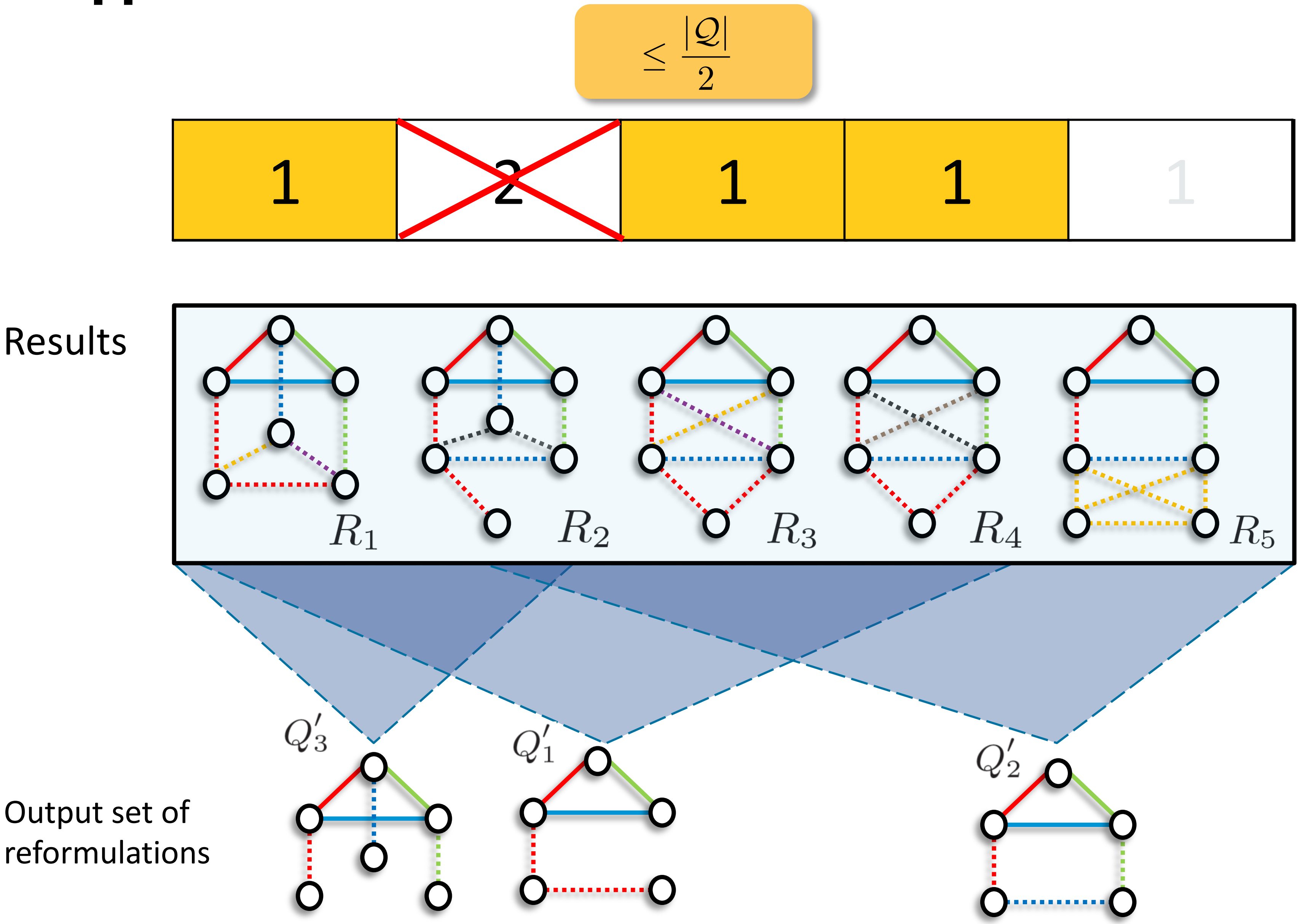
For a reformulation $Q' \in \mathbb{S} \setminus \mathcal{Q}$ it holds that

$$\max_{Q'' \in \mathcal{T}_{Q'}} \Delta_f(\mathcal{Q}, Q'') \leq \overline{\Delta_f}(\mathcal{Q}, Q') =$$

$$= \frac{1}{2} \vec{u}_{\mathcal{Q}} \cdot \vec{x}_{Q^*} + \lambda \left(\|\vec{m}_{\mathcal{Q}}\| + |\mathcal{Q}| \times \|\vec{x}_{Q^*}\| - 2 \vec{m}_{\mathcal{Q}} \cdot \vec{x}_{Q^*} \right).$$

db Trento

Upper bound



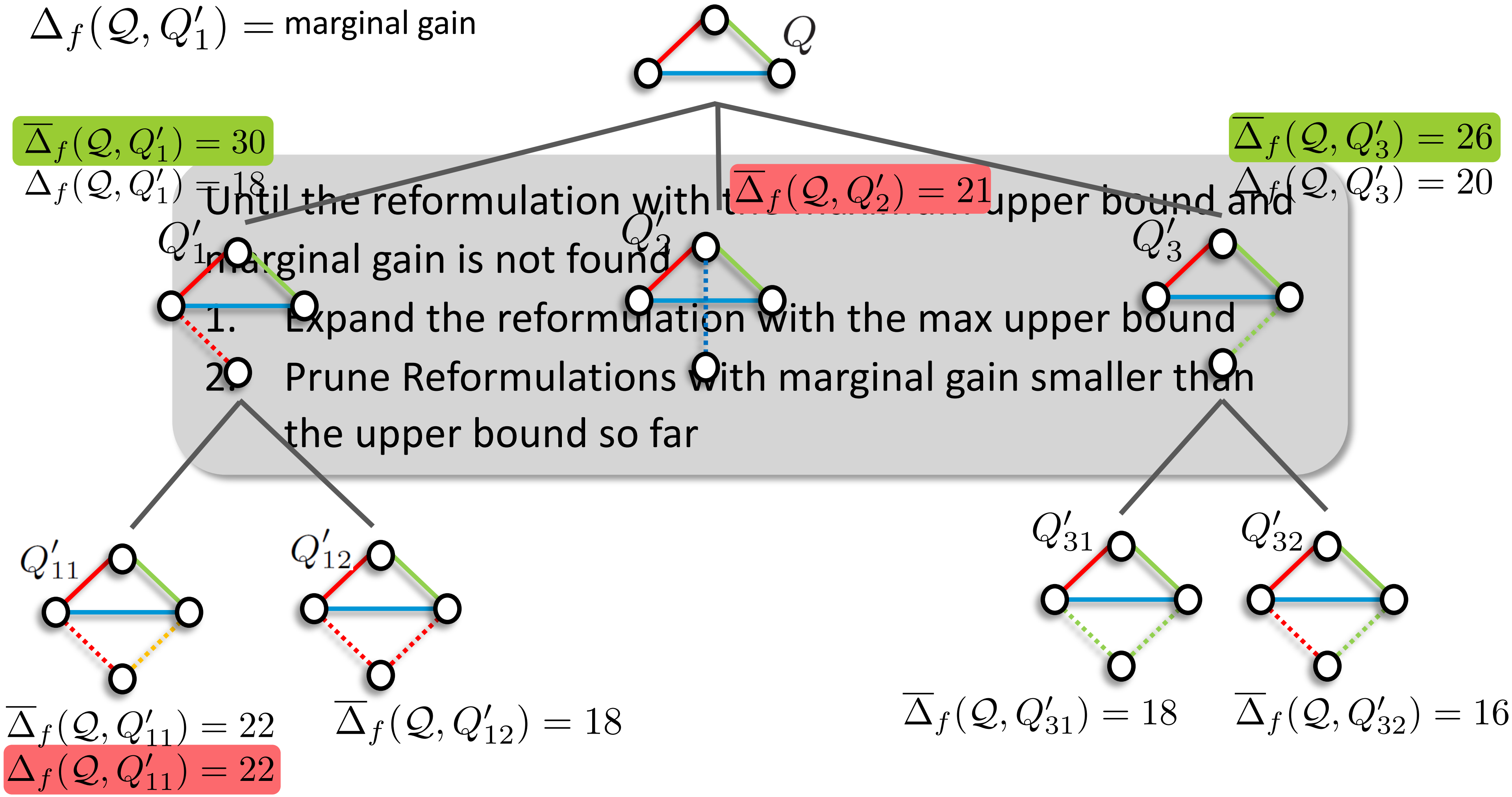
$\leq \frac{|\mathcal{Q}|}{2}$
1
2
1
1
1
Results
R_1
R_2
R_3
R_4
R_5
Q'_3
Q'_1
Q'_2
Output set of reformulations

The Fast_MMPG Algorithm

$\overline{\Delta}_f(\mathcal{Q}, Q'_1)$ = upper bound
$\Delta_f(\mathcal{Q}, Q'_1)$ = marginal gain

Until the reformulation with the maximum upper bound and marginal gain is not found

1. Expand the reformulation with the max upper bound
2. Prune Reformulations with marginal gain smaller than the upper bound so far



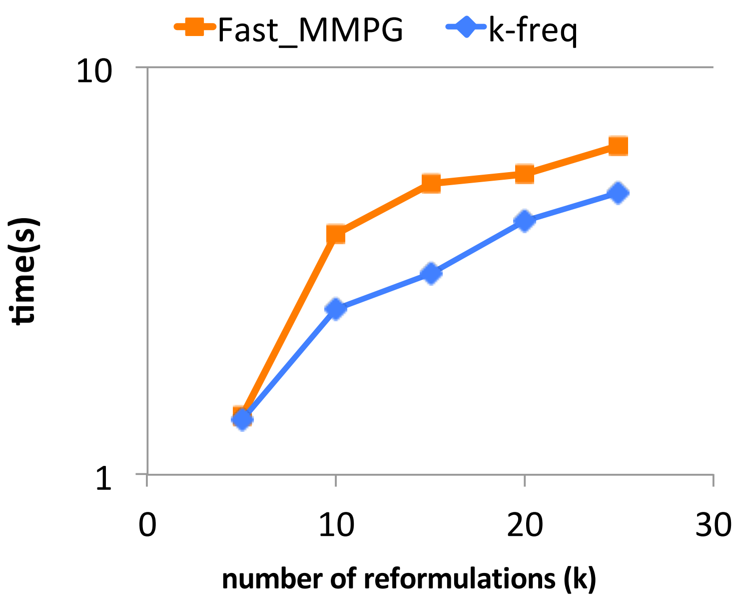
db Trento

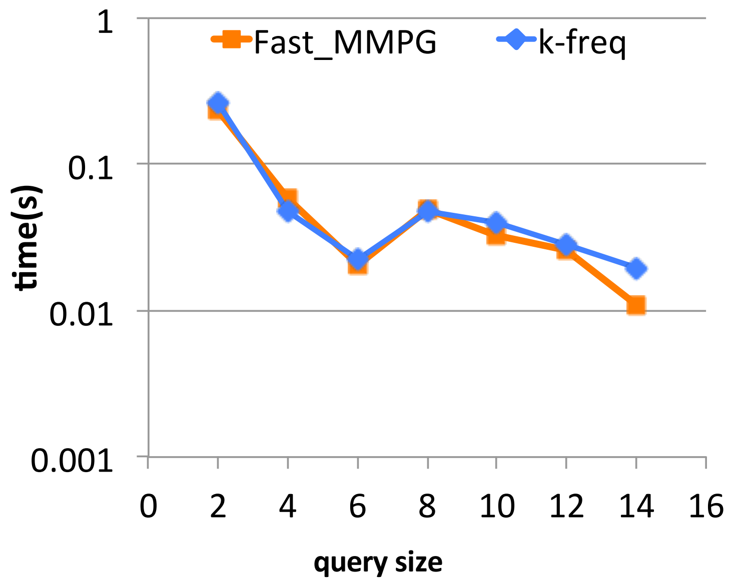
Experimental Setup

- **Datasets:**
  - AIDS: 10k chemical compounds
  - Financial: 17k transaction workflows
  - Web: 13k interactions with a recommender system
- **Baseline algorithms:**
  - k-freq: returns top-k frequent supergraphs of a query
  - LIndex: informative patterns index
- **Experiments:**
  - Time and objective function value varying k, query size, λ
  - Anecdotal
  - Scalability

dbTrento

Time Comparison





Number of reformulations

1. k-freq runs only slighly faster
2. Time increases linearly in k
3. Fast_MMPG has real-time performance

Query size

1. Fast_MMPG comparable to k-freq
2. Time decreases with query size (less reformulations)

Objective function gain

	λ				
	0	0.01	0.05	0.1	0.5
Fast_MMPG	433	613	1 345	2 260	9 566
k-freq	409	540	1 063	1718	6 954
gain (%)	6	12	21	24	27

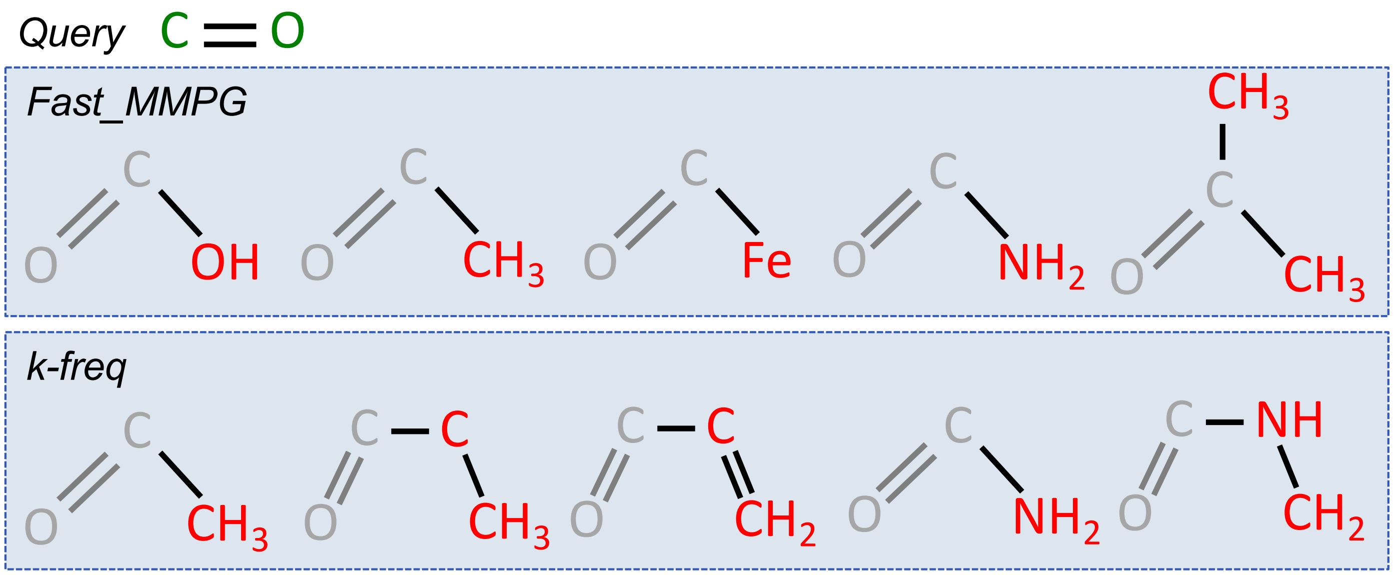
Analysis

1. Lambda correctly moves the objective function towards diversity
2. k-freq only captures coverage

$$f(\mathcal{Q}) = cov(\mathcal{Q}) + \lambda \sum_{Q',Q'' \in \mathcal{Q}} div(Q', Q'')$$

db Trento

Qualitative evaluation

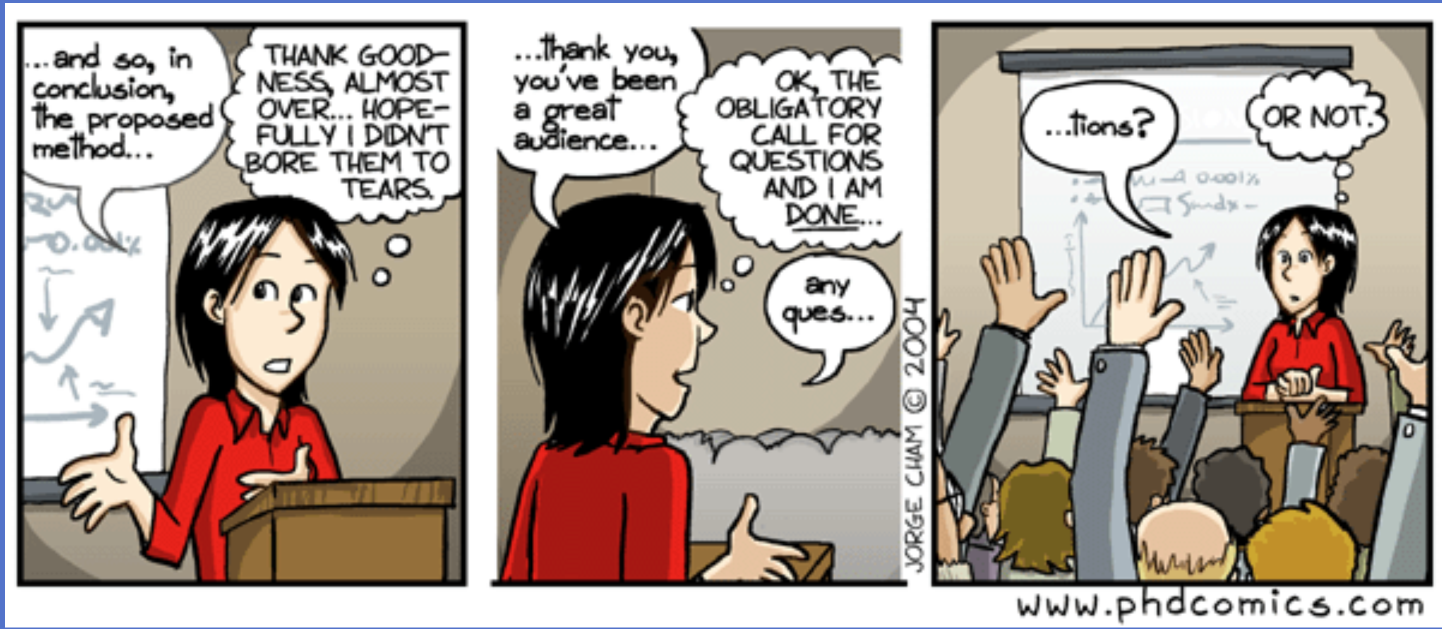


Analysis

- k-freq finds reformulation of the same superquery
- Fast_MMPG returns reformulations with more diversified structures

Conclusions

- First study of the problem in **graph databases**
- **Principled** objective function optimizing **coverage** and **diversity**
- Algorithmic solutions **with quality guarantees** and **real time responses**



Thank you!

Questions?

dbTrento