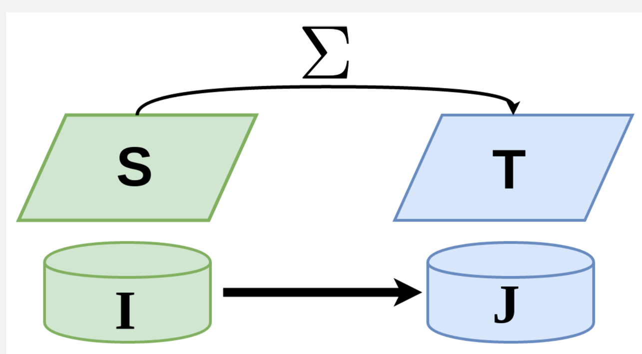
ACTIVE LEARNING OF GAV SCHEMA MAPPINGS

[♤]Balder ten Cate, [◊,♧]Phokion G. Kolaitis, [◊]Kun Qian, and [♡]Wang-Chiew Tan
[♤]Google Inc. [◊]IBM Research – Almaden [♧]UCSC [♡]Megagon Labs

PODS 2018
Houston, TX
USA

1 Schema Mappings and Data Exchange

- Schema mapping: triple $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ with Σ constraints between S and T

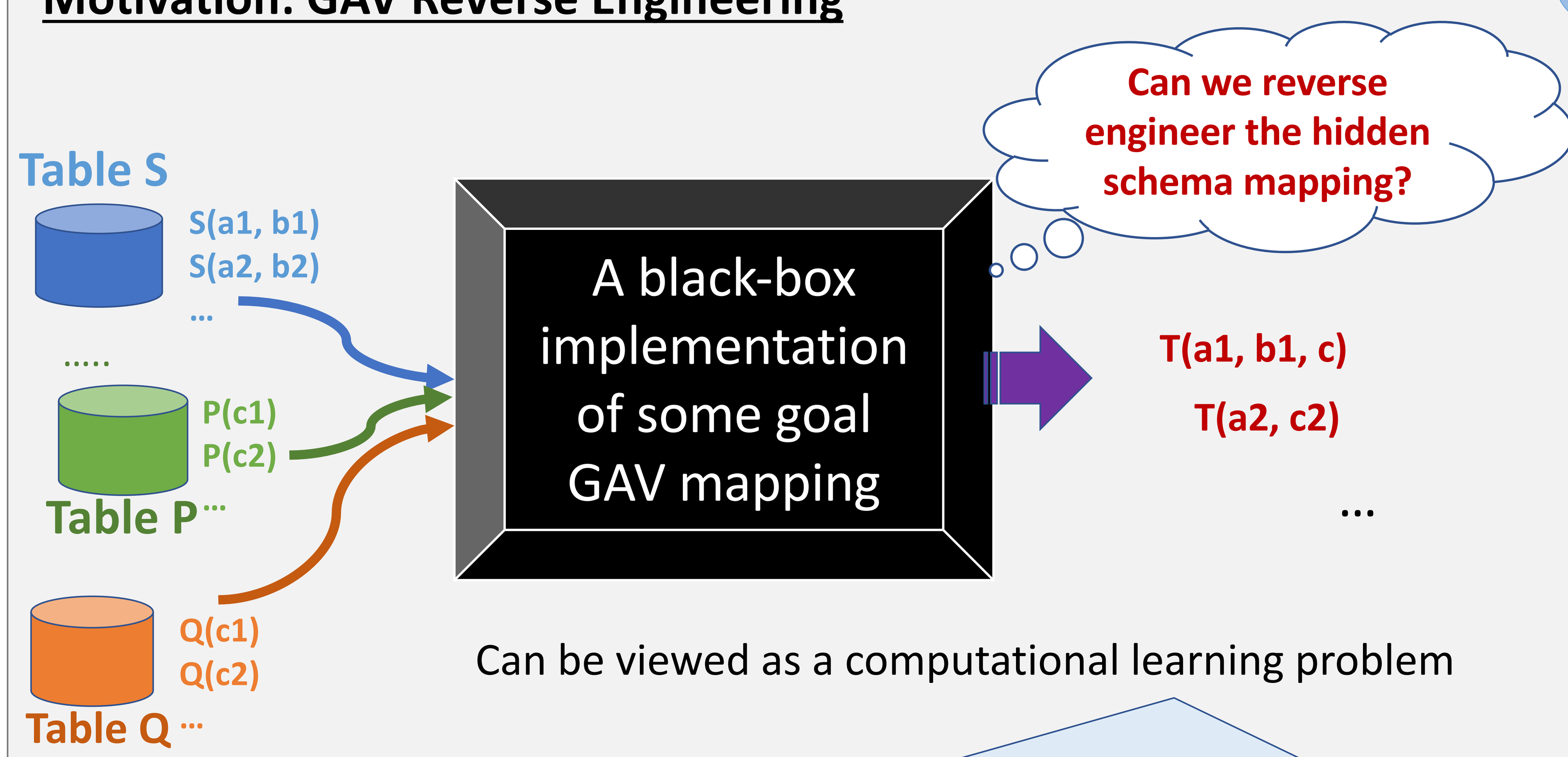


- Data exchange problem: given an S-instance **I**, find a T-instance **J** so that (I,J) satisfies Σ.
- Schema-mapping language: GAV (Global-As-View) constraints

$$\forall \mathbf{x}(\varphi(\mathbf{x} \to T(\mathbf{x}))$$

$\text{Example}: \forall v \forall u(\text{Node}(v) \wedge \text{Node}(u) \to \text{Edge}(v,u))$

2 Motivation: GAV Reverse Engineering



Can be viewed as a computational learning problem

- **Angluin's Learning Model**: Identify an unknown "goal concept" by asking queries
- Typical queries asked to an oracle:
  - membership/labeling queries
  - equivalence queries

3 Learnability of GAV Mappings

- **Theorem ([ten Cate et al., 2013]).** GAV mappings are efficiently learnable with equivalence and labeling queries.
  - ***Labeling oracle***: given a source instance I, return the canonical universal solution J for I.
  - ***Equivalence oracle:*** given two GAV mappings M and M^*, check if M and M^* are logically equivalent; if not, a counter-example is returned.
- **Fact.** The ExactGAV algorithm from [ten Cate et al., 2013] is hard to implement
  - A black-box implementation can serve as the labeling oracle.
  - However, an equivalence oracle may not be available in practice.

4 GAVLearn: An Active Learning Algorithm

- Adapted from the ExactGAV algorithm
- **Main idea**: replace the equivalence oracle with **conformance testing**

Input: **G** – a black-box implementation of the goal GAV mapping
E – a set of universal examples for G
Output: **H** – a GAV mapping that fits E whose size is at most the size of G

- **Conformance testing**: Use **E** to check if the behavior of H on E conforms to the specification of **G**
  - Accept H if it agrees with G on every example $(I, J) \in E$.
  - Otherwise, there is an example $(I, J) \in E$, on which H and G disagree
  - Use counter-example to actively do experiments

5 Example

$\mathrm{M}(x,y) \wedge \mathrm{N}(y,z) \to \mathrm{Q}(x,y,z)$
$\mathrm{S}(x,y) \wedge \mathrm{R}(y,z) \to \mathrm{T}(x,z)$

Goal Mapping - G

(I,J)

Examples - E

$\mathrm{I} = \{S(a,b), R(b,c), M(a,b), N(b,c)\},$
$\mathrm{J} = \{T(a,c), Q(a,b,c)\}$

Many mappings can perfectly describe the semantics of (I,J)

$\{\mathrm{M}(x,y) \wedge \mathrm{N}(y,z) \to \mathrm{T}(x,z), \mathrm{S}(x,y) \wedge \mathrm{R}(y,z) \to \mathrm{Q}(x,y,z)\}$
$\{\mathrm{M}(x,y) \wedge \mathrm{R}(y,z) \to \mathrm{T}(x,z), \mathrm{S}(x,y) \wedge \mathrm{N}(y,z) \to \mathrm{Q}(x,y,z)\}$
...

Run GAVLearn on G and E

1st iteration

1(a) - Initialize an H = { }. We have that H and G disagree on (I,J). Choose T(a,c) from J and form a counter-example (I,{T(a,c)})

1(b) Active learning

$\mathrm{I} \to \{\cancel{S(a,b)}, R(b,c), M(a,b), N(b,c)\} \xrightarrow{\text{Apply G}} \{Q(a,b,c)\}$ **Keep S(a,b)**
$\mathrm{I} \to \{S(a,b), \cancel{R(b,c)}, M(a,b), N(b,c)\} \xrightarrow{\text{Apply G}} \{Q(a,b,c)\}$ **Keep R(b,c)**
$\mathrm{I} \to \{S(a,b), R(b,c), \cancel{M(a,b)}, N(b,c)\} \xrightarrow{\text{Apply G}} \{T(a,c)\}$ **Remove M(a,b)**
$\mathrm{I} \to \{S(a,b), R(b,c), \cancel{M(a,b), N(b,c)}\} \xrightarrow{\text{Apply G}} \{T(a,c)\}$ **Remove N(b,c)**

1(c) - Construct a GAV constraints from {S(a,b), R(b,c)} and T(a,c) and add it to H

$$\mathrm{H} = \{\mathrm{S}(x,y) \wedge \mathrm{R}(y,z) \to \mathrm{T}(x,z)\}$$

2nd iteration

2(a) – H and G disagree on (I,J). By chasing I with H, we obtain T(a,c) but miss Q(a,b,c). Choose Q(a,b,c) and form a new counter-example (I, {Q(a,b,c)})

2(b) Repeat the active learning process in 1(b) to obtain I = {M(a,b), N(b,c)} and F'=Q(a,b,c)

2(c) – Construct a GAV constraint from {M(a,b), N(b,c)} and Q(a,b,c) and add it to H

$$\mathrm{H} = \{\mathrm{S}(x,y) \wedge \mathrm{R}(y,z) \to \mathrm{T}(x,z), \mathrm{M}(x,y) \wedge \mathrm{N}(y,z) \to \mathrm{Q}(x,y,z)\} = \mathrm{G}$$

6 Results - Part I: Theoretical Guarantees

Theorem. GAVLearn is an Occam algorithm for GAV mappings.

Occam learnability implies PAC learnability (Blumer et al., 1987)

Corollary. GAVLearn is a PAC algorithm for GAV mappings.

Results - Part II: Experiments

- Use iBench (Arocena et al., 2015) to generate:
  - SIMPLE, MODERATE, and COMPLEX mapping scenarios
  - Data examples for each scenario (split 50-50 into training and testing)

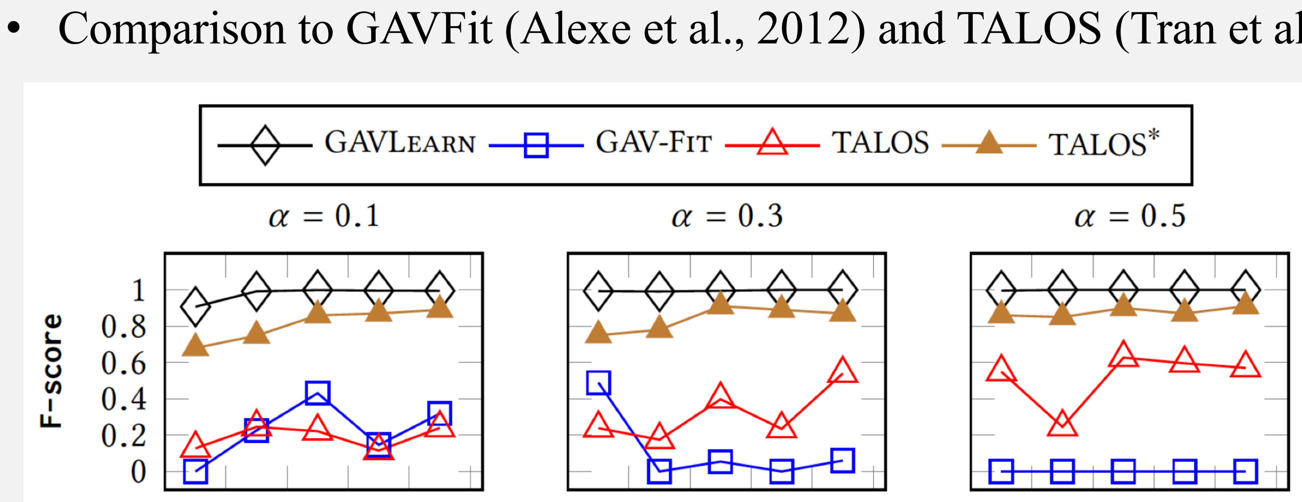
Table 4: Results of GAVLearn on complex type

α	n	$\lvert E\rvert$	$\overline{Comp}$	$\overline{Rep}$	$\overline{Recall}$	F_s	$\overline{\text{Time}}$
0.1	10	5	0.53±5%	0.87±10%	0.882±10%	0.937	17.8s
	30	15	0.60±0%	1	1	1	20.7s
	50	25					17.6s
	70	35					22.4s
	90	45					19.7s
0.3	10	5	0.60±0%	1	1	1	1m26s
	30	15	0.60±1%				1m35s
	50	25	0.60±0%				1m34s
	70	35	0.60±1%				1m33s
	90	45	0.60±0%	0.98±2%	0.999	0.999	1m45s
0.5	10	5	0.63±3%	0.98±2%	0.998±4‰	0.999	4m15s
	30	15	0.64±4%	0.92±5%	0.997±2‰	0.998	4m43s
	50	25	0.69±3%	0.92±4%	0.998±1%	0.999	6m55s
	70	35	0.73±4%	0.87±9%	0.998±1%	0.999	5m44s
	90	45	0.74±2%	0.88±8%	0.998±1%	0.999	10m9s

F-scores

In all cases, F-scores are above 90% (many of them are 100%)

- Comparison to GAVFit (Alexe et al., 2012) and TALOS (Tran et al., 2014)



GAVLearn outperforms GAVFit ant TALOS

Knowledge Refinement via Rule Selection (AAAI-19)

[♧,◊] Phokion G. Kolaitis, [◊]Lucian Popa, and [◊] Kun Qian
[♧] UC Santa Cruz, [◊] IBM Research – Almaden

Motivation

- Rules (Horn formulas) are ubiquitous in AI

∀x,y,z (PARENT(x,z) ∧ PARENT(y,z) → SIBLING(x,y))
SIBLING(x,y) :- PARENT(x,z), PARENT(y,z)

Association rules

TID	Items
1	Cake, Milk
2	Cake, Diaper, Beer
3	Diaper, Beer, Coke
4	Beef, Diaper, Beer
5	Diaper, Bread, Milk, Coke

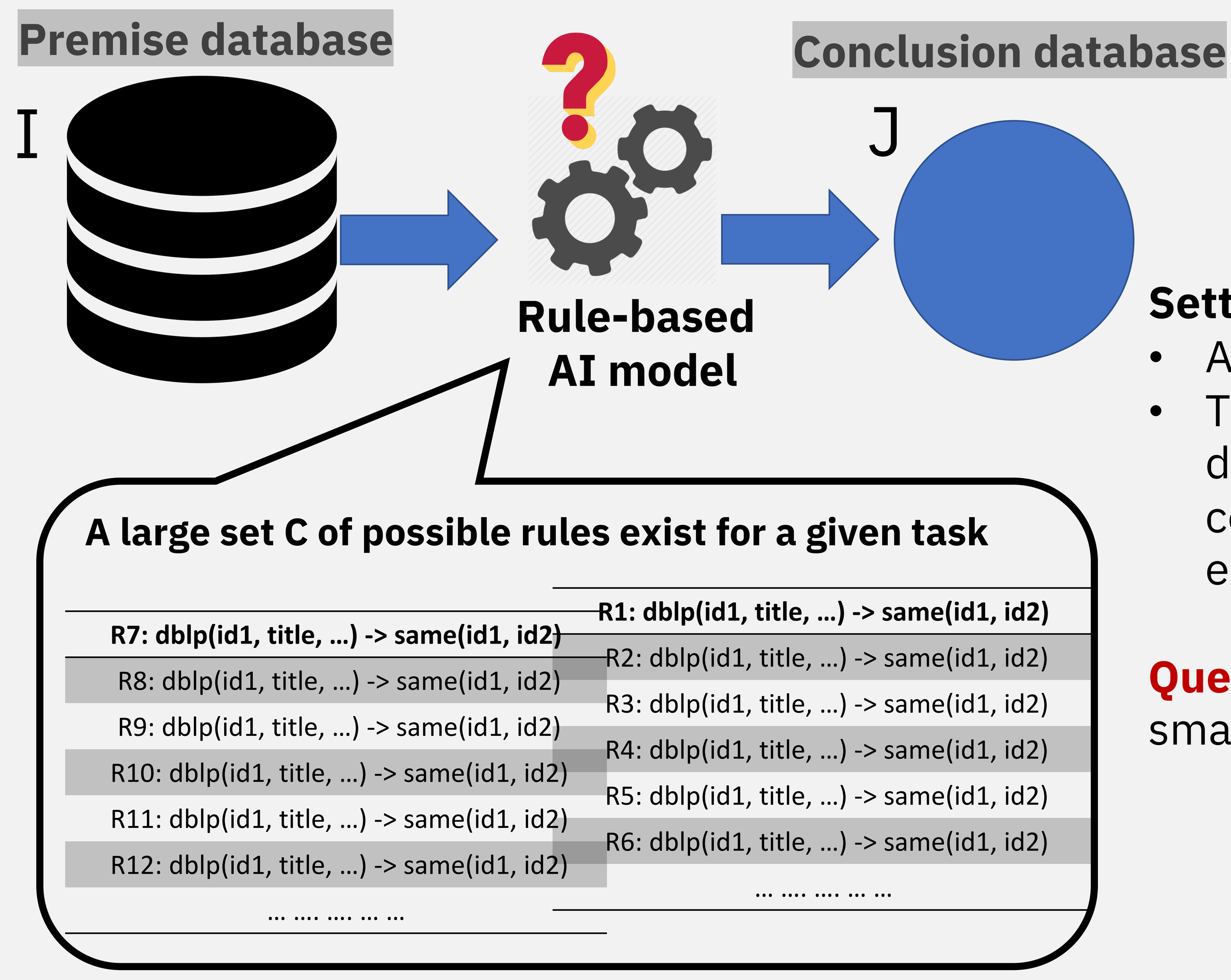
→

{Cake} -> {Milk}
{Cake, Diaper} -> {Beer}
{Diaper} -> {Beer}
{Beef, Diaper} -> {Beer}
…… …

Entity resolution rules

Matching publications

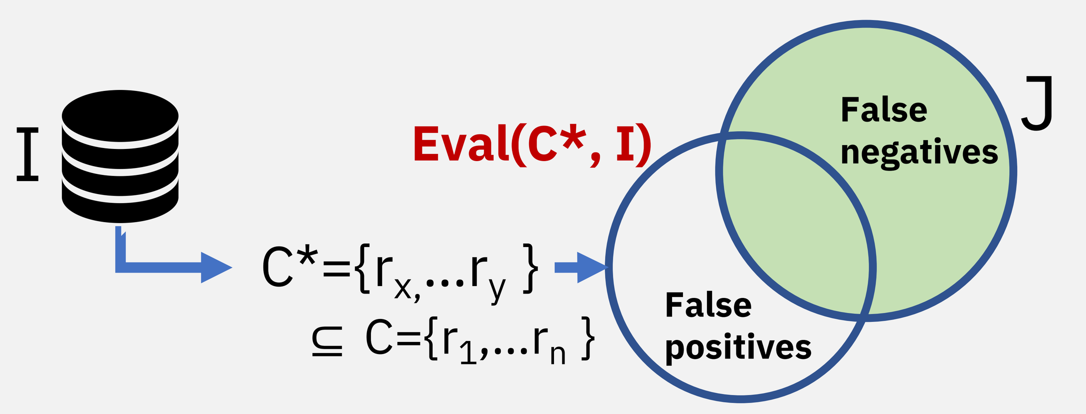
DBLP(id, **title, authors, venue,** year)
∧ ACM(id', **title, authors, venue,** year') → Same(id,id')



Setting:
- A large set C of possible rules is available.
- The actual semantics of the task is given via data: a premise database I together with a conclusion database J that represents the expected output of the task on input I.

Question: Can we capture the semantics via a small subset of C?

Min Rule Selection Problem



- **Total error** = false positives + false negatives

Definition. Min Rule-Select Problem

Given an input-ouput databases (I, J) and a set C of rules, compute a subset C* of C such that the total error of Eval(C*, I) w.r.t. J is minimized.

Theorem 1

Min Rule-Select is an NP-hard optimization problem.

Unless NP=P, there is no polynomial time algorithm that solves the problem exactly.

Are there PTIME approximation algorithms with formal guarantees?

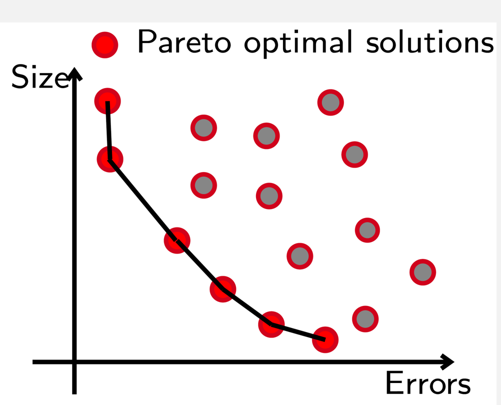
Theorem 2

Min Rule-Select is approximable within a factor of $2\sqrt{|C| + |J| \cdot \log|J|}$, where $|C|$ is the number of input rules and $|J|$ is the size of the conclusion instance.

We show that Min Rule-Select is "equivalent" to the Positive-Negative Partial Set-Cover Problem [Miettinen, 2008]

Bi-objective and Bi-level Optimization

What if we want to optimize both the *error* and the *size* of the set of rules (*error* and *size* are incomparable quantities).



Definition. Pareto Optimal Solution

Given a set C of rules and an input-output databases (I,J), a subset C* of C is a ***Pareto-Optimal*** solution if there is no subset C' of C such that size(C') < size(C*) and error(C')<error(C*)

Definition. Bi-level Optimal Solution

Given a set C of rules and an input-output databases (I,J), a subset C* of C is a ***Bi-level Optimal*** solution if it has minimum error and also has minimum size among all minimum-error solutions.

Theorem 3

The following problems are coNP-complete: given an input-output database (I,J) and a C of rules, and given subset C* of C:

- ***is C* a Pareto optimal solution?***
- ***is C* a bi-level optimal solution?***

Summary of Results and Future Work

		False Positive Errors	False Positive + False Negative Errors
Rule-Select		NP-complete	NP-complete
Exact Rule-Select		DP-complete	DP-complete
Min Rule-Select	approx. upper bound	$2\sqrt{\lvert\mathcal{C}\rvert \log \lvert J\rvert}$	$2\sqrt{(\lvert\mathcal{C}\rvert + \lvert J\rvert)\log \lvert J\rvert}$
	approx. lower bound	$2^{\log^{1-\epsilon}(\lvert\mathcal{C}\rvert)}$, for every $\epsilon > 0$	$2^{\log^{1-\epsilon}(\lvert J\rvert)}$, for every $\epsilon > 0$
Pareto Optimal Solution		coNP-complete	coNP-complete
Pareto Front Membership		DP-complete	DP-complete
Bi-level Optimal Solution		coNP-complete	coNP-complete
Bi-level Optimal Value		DP-complete	DP-complete

Directions for future work

- Approximating the Pareto front of the rule selection problem.
- Experimental evaluation.