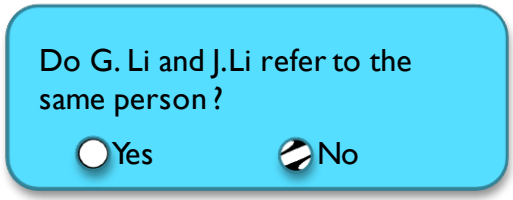
CDB: A Crowd-Powered Database System

Guoliang Li, Chengliang Chai, Ju Fan, Xueping Weng, Jian Li, Yudian Zheng Yuanbing Li, Xiang Yu, Xiaohang Zhang, Haitao Yuan

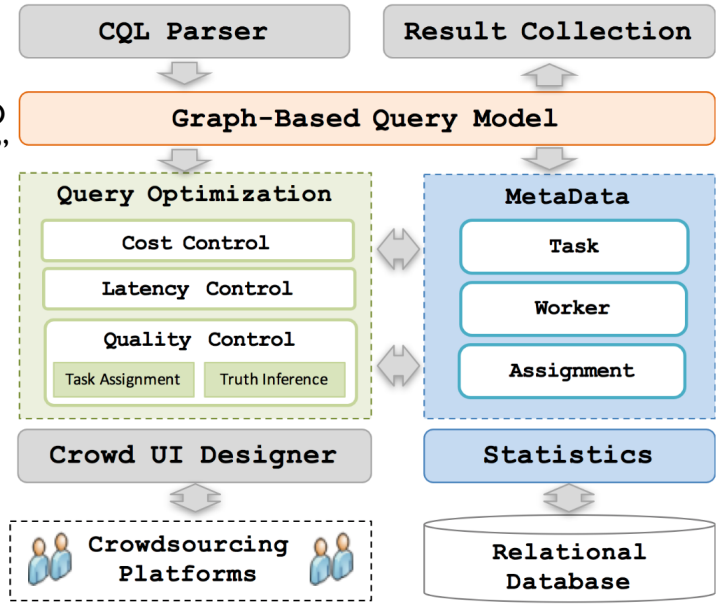
Crowd-Based Database

Crowd-Based database can execute some queries which are hard for traditional database

Select Affiliation From Professor, Paper
Where Professor CROWDJOIN Paper AND
Paper.Conf. CROWDEQUAL "SIGMOD"



.....



Affiliation
Tsinghua
Peking Univ.

Professor

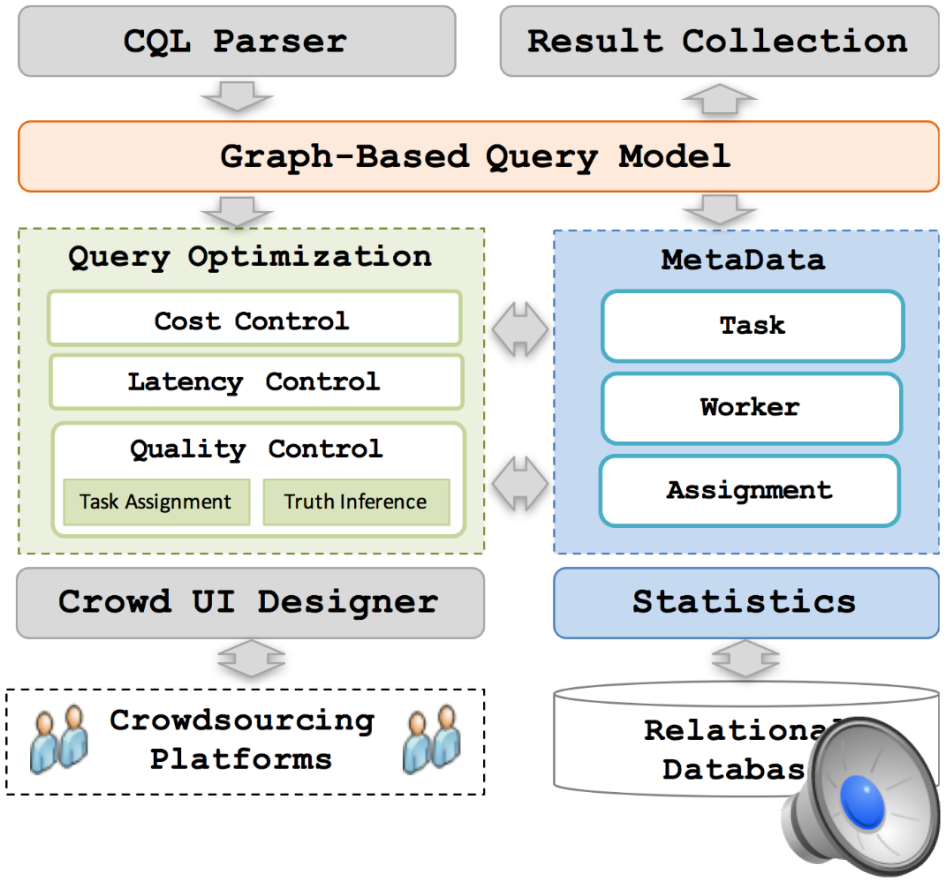
Prof.	Affiliation
Guo. Li	Tsinghua
J. Li	Peking Univ.

Paper

Author	Title	Conf.
G. Li	xxx	SIGM
Jian. Li	xxx	SIG

Workflow

- A requester submits her query using CQL, which will be parsed by CQL Parser.
- Graph-based query model builds a graph model based on the parsed result.
- Query optimization generates an optimized query plan
- Crowd UI Designer designs various interfaces and interacts with underlying crowdsourcing plat- forms.



Motivation

- Optimization Models

  Existing works: Tree-based model (table-level)

  CDB: Graph-based model (tuple-level).

- Optimizing Goals:

  Existing works: Mainly on cost.

  CDB: Focus on multiple goals (cost, quality and latency).

Graph Model

Weight(Jaccard, ED) $w(e)$ $>threshold$

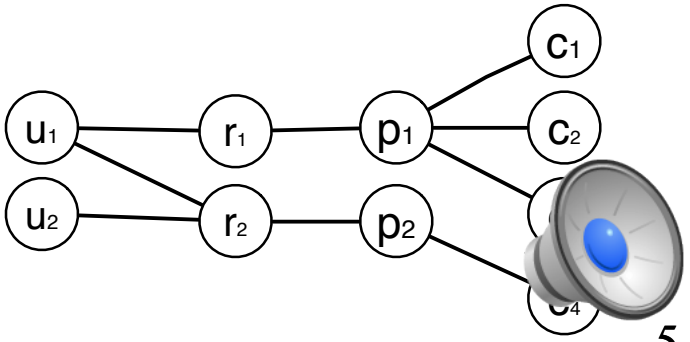
	Country	Name
u_1	UK	Univ. of Cambridge
u_2	US	Microsoft

	Affiliation	Name
r_1	University of Cambridge	Nandan Parameswaran
r_2	Microsoft Cambridge	S. Chaudhuri

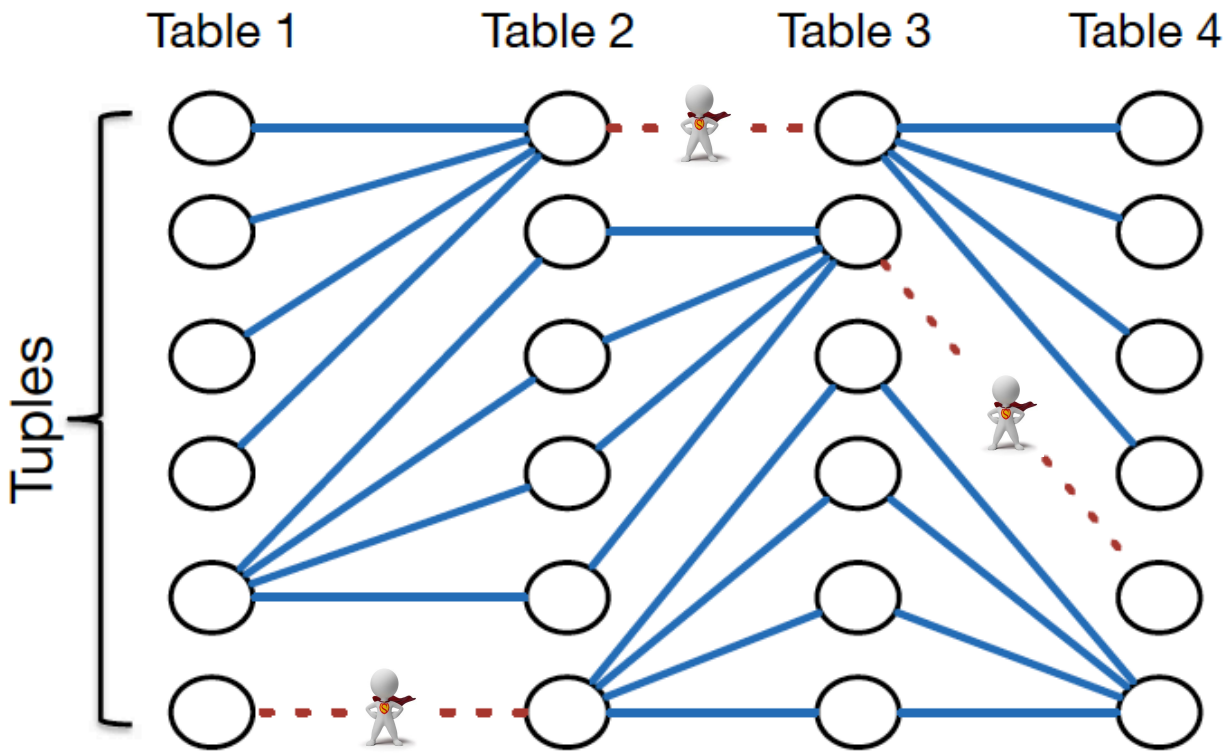
	Number	Title
c_1	16	DataSift: An Expressive and Accurate Crowd-Powered Search Toolkit.
c_2	4	A crowd powered search toolkit
c_3	0	A Crowd Powered System for Similarity Search
c_4	1	Query portals: dynamically generating portals for entity-oriented web queries.

	Title	Author
p_1	DataSift: a crowd-powered search toolkit	Aditya G. Parameswaran
p_2	Dynamically generating portals for entity-oriented web queries.	Surajit Chaudhuri

- For each table T in the CQL query, there is a vertex for each tuple in this table.
- For each crowd join predicate $T.C_i$ CROWDJOIN $T'.C_i$ in the CQL query, there is an edge e between $t \in T$ and $t' \in T'$ with $w(e)$ $>threshold$.



Tuple-level VS Table-level



9 questions + 5 questions + 1 question = 15 questions

3 questions

Differences with Existing Systems

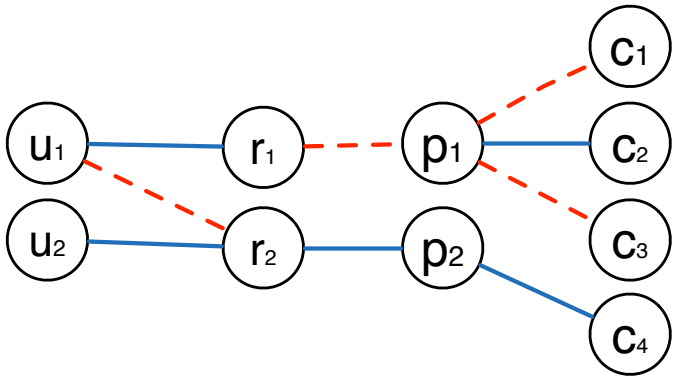
Reduce

		CrowdDB	Qurk	Deco	CrowdOP	CDB
Optimized Crowd Operators	COLLECT	√	×	√	×	√
	FILL	√	×	√	√	√
	SELECT	√	√	√	√	√
	JOIN	√	√	√	√	√
Optimization Objectives	Cost	√	√	√	√	√
	Latency	×	×	×	√	√
	Quality	MV	MV	MV	MV	√
Optimization Strategies	Cost-Model	×	√	√	√	√
	Tuple-Level	×	×	×	×	√
	Budget-Supported	×	×	×	×	√
Task Deployment	Cross-Market	×	×	×	×	√

Contributions

- Optimization Models: Graph-based model (tuple-level).
- Optimizing Goals: Focus on multiple goals (cost, quality and latency).
- Many commonly used crowd-powered operators.
- Cross-market HITs deployment.

Cost Control



CQL Query Candidate:

(u_1, r_1, p_1, c_1), (u_1, r_1, p_1, c_2), (u_1, r_1, p_1, c_3), (u_1, r_2, p_2, c_4), (u_2, r_2, p_2, c_4)

CQL Query Answer:

(u_2, r_2, p_2, c_4)

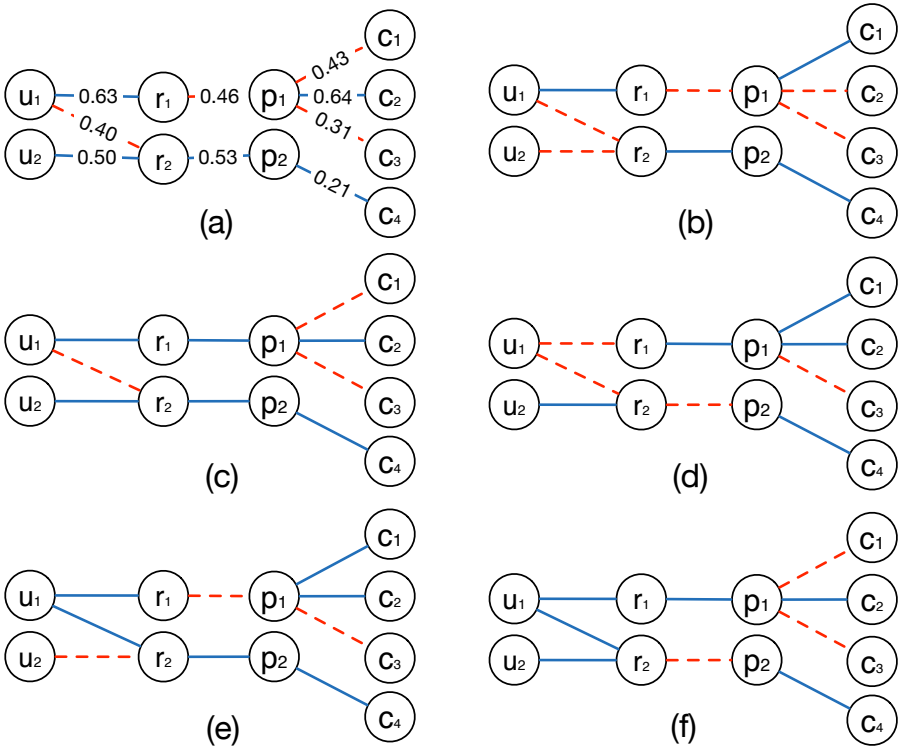
Given the colors of every edge, how to select he minimum number of edges to find al the answers ?

Min-Cut Based Algorithm (refer to the paper for detail)

In the example, the optimal edges are (u_2, r_2) (r_2, p_2) (p_2, c_4) (r_1, p_1) (u_1, r_2).

Consider the case where the colors of edges are **unknown**. We aim to ask fewer edges to find all answers with high probability.

Sample Average



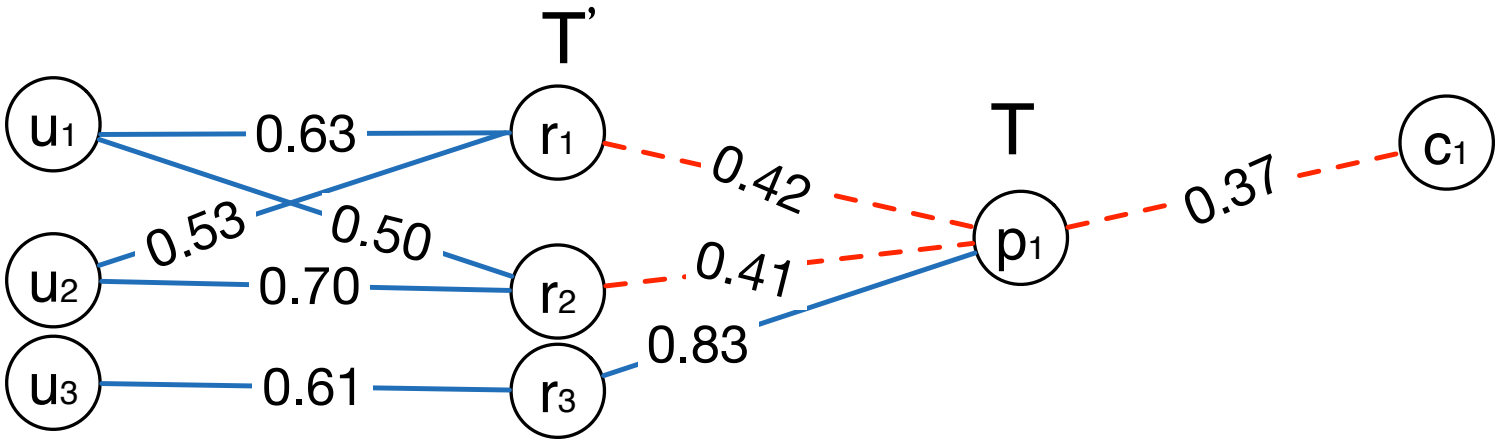
Given *S* sample graphs, select the minimum number of edges to resolve all samples

(b) (u_1, r_2) (u_2, r_2) (r_1, p_1)
(d) (u_1, r_2) (u_2, r_2) (r_2, p_2)
(e) (r_1, p_1) (u_2, r_2) (u_1, r_2) (u_2, r_2) (r_2, p_2) (p_2, c_4)
……

NP-HARD

Greedy algorithm

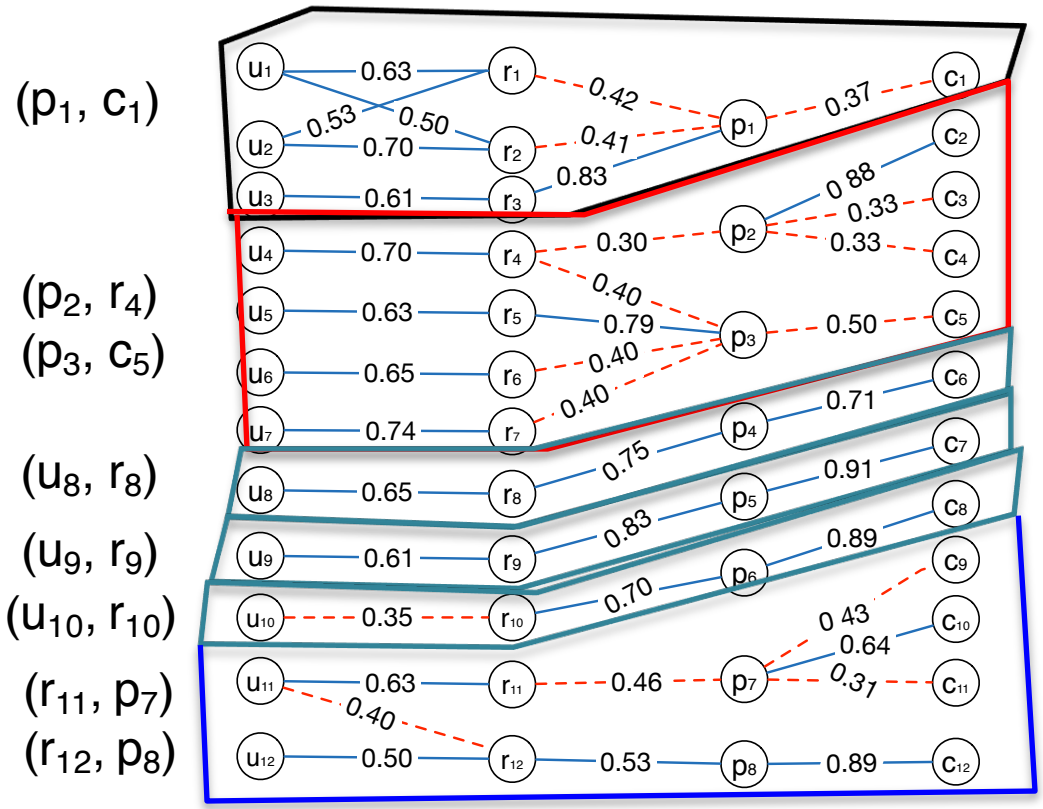
Expectation-based Method



$$\mathbb{E}(t,t') = \frac{\prod_{i=1}^{x}(1-\omega(t,t_i))}{x}\alpha + \frac{\prod_{i=1}^{y}(1-\omega(t_i,t'))}{y}\beta.$$

$E(r_1,p_1)=(1-0.42)*2 + (1-0.42)*(1-0.41)*(1-0.83)*6/3 =$ 1.37

Latency Control



Connected Components

e.g. (p_1, c_1) (p_2, c_2)

Edges Containing Tuples from the Same Table.

e.g. (p_1, r_1) (p_1, r_2)

Quality Control

Truth Inference

W={*w*}: a set of workers

T={*t*} : a set of tasks

Vt={*(w,a)*}: worker *w* provides answer *a* for task *t*

The probability of the *i*-th choice being the truth for task t is computed as:

$$p_i = \frac{\prod_{(w,a)\in V_t} (q_w)^{\mathbb{1}\{i=a\}} \cdot \left(\frac{1-q_w}{\ell-1}\right)^{\mathbb{1}\{i\neq a\}}}{\sum_{j=1}^{\ell} \prod_{(w,a)\in V_t} (q_w)^{\mathbb{1}\{j=a\}} \cdot \left(\frac{1-q_w}{\ell-1}\right)^{\mathbb{1}\{j\neq a\}}}$$

Other types of tasks: refer to the paper for detail

Quality Control

Task Assignment

Assign a set of *k* tasks to worker *w*, such that the quality can be improved the most.

Two main problems:

(i) unknown ground truth

(ii) how the worker can answer each task.

Distribution of choices being true for each task t

$$P=(p_1, p_2, \ldots p_{l-1})$$

Entropy function:

$$H(P)=-\sum p_i \log(p_i)$$

The lower H(p) is, the more consistent P is, the higher quality will be achieved.

Quality Control

Task Assignment

Probability that the *i*-th choice will be answered by *w* :

$$p_i \cdot q_w + (1 - p_i) \cdot \frac{1 - q_w}{\ell - 1}$$

Then after worker *w* answers task *t* with the *i*-th choice, the distribution is as follows :

$$\vec{p'} = \left(\frac{p_1 \cdot \frac{1-q_w}{\ell-1}}{\Delta}, \ldots, \frac{p_i \cdot q_w}{\Delta}, \ldots, \frac{p_\ell \cdot \frac{1-q_w}{\ell-1}}{\Delta} \right)$$

The expected quality of improvement

$$\mathcal{I}(t) = \mathcal{H}(\vec{p}) - \sum_{i=1}^{\ell} \left[p_i \cdot q_w + (1 - p_i) \cdot \frac{1 - q_w}{\ell - 1} \right] \cdot \mathcal{H}(\vec{p'})$$

Other types of tasks: refer to the paper for detail

Task Type & UI Designer

Please choose the brand of the phone

- (●) Apple
- ( ) Samsung
- ( ) Blackberry
- ( ) Other

Which ones are correct?

- [ ] The same band
- [ ] The same size
- [x] Different bands
- [x] Different sizes

Please fill the attributes of the product

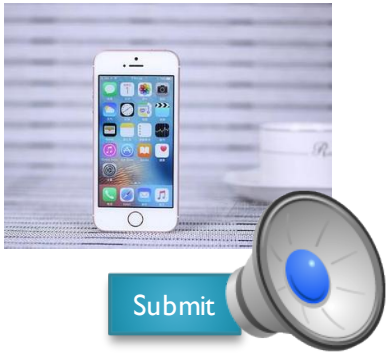
Brand _____

Price _____

Size _____

Whether has camera _____

Please submit a picture of a phone, which is the same brand as the left one.



Experiment

Dataset

Paper

Table	#Records	Attributes
Paper	676	**author**,*title*,conference
Citation	1239	*title*, number
Researcher	911	affiliation, **name**, gender
University	830	name, city, country

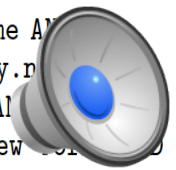
Award

Table	#Records	Attributes
Celebrity	1498	**name**, *birthplace*, birthday
City	3220	*birthplace*, country
Winner	2669	**name**, award
Award	1192	name, place

CQL Queries

Table 4: The 5 representative queries used on paper and award.

Query	Dataset paper	Dataset award
2 Joins (2J)	SELECT Paper.title,Researcher.affiliation, Citation.number FROM Paper, Citation, Researcher WHERE Paper.title CROWDJOIN Citation.title AND Paper.author CROWDJOIN Researcher.name	SELECT Winner.award, City.country FROM Winner, City, Celebrity WHERE Celebrity.name CROWDJOIN Winner.name AND Celebrity.birthplace CROWDJOIN City.name
2 Joins 1 Selection (2J1S)	SELECT Paper.title,Researcher.affiliation, Citation.number FROM Paper, Citation, Researcher WHERE Paper.title CROWDJOIN Citation.title AND Paper.author CROWDJOIN Researcher.name AND Paper.conference CROWDEQUAL "sigmod"	SELECT Winner.award, City.country FROM Winner, City, Celebrity WHERE Celebrity.name CROWDJOIN Winner.name AND Celebrity.birthplace CROWDJOIN City.name AND Celebrity.birthplace CROWDEQUAL "New York"
3 Joins (3J)	SELECT Paper.title,Citation.number,University.country FROM Paper, Citation, Researcher,University WHERE Paper.title CROWDJOIN Citation.title AND Paper.author CROWDJOIN Researcher.name AND University.name CROWDJOIN Researcher.affiliation	SELECT Winner.name, Award.place, City.country FROM Winner, City, Celebrity, Award WHERE Celebrity.name CROWDJOIN Winner.name AND Celebrity.birthplace CROWDJOIN City.name AND Winner.award CROWDJOIN Award.name
3 Joins 1 Selection (3J1S)	SELECT Paper.title,Citation.number FROM Paper, Citation, Researcher,University WHERE Paper.title CROWDJOIN Citation.title AND Paper.author CROWDJOIN Researcher.name AND University.name CROWDJOIN Researcher.affiliation AND University.country CROWDEQUAL "USA"	SELECT Winner.name,City.country FROM Winner, City, Celebrity, Award WHERE Celebrity.name CROWDJOIN Winner.name AND Celebrity.birthplace CROWDJOIN City.name AND Winner.award CROWDJOIN Award.name AND Award.place CROWDEQUAL "US"
3 Joins 2 Selections (3J2S)	SELECT Paper.title,Citation.number FROM Paper, Citation, Researcher,University WHERE Paper.title CROWDJOIN Citation.title AND Paper.author CROWDJOIN Researcher.name AND University.name CROWDJOIN Researcher.affiliation AND Paper.conference CROWDEQUAL "sigmod" AND University.country CROWDEQUAL "USA"	SELECT Winner.name,City.country FROM Winner, City, Celebrity, Award WHERE Celebrity.name CROWDJOIN Winner.name A Celebrity.birthplace CROWDJOIN City.n Winner.award CROWDJOIN Award.name A Celebrity.birthplace CROWDEQUAL "New Award.place CROWDEQUAL "US"



Cost

Reduce 2-3 times cost

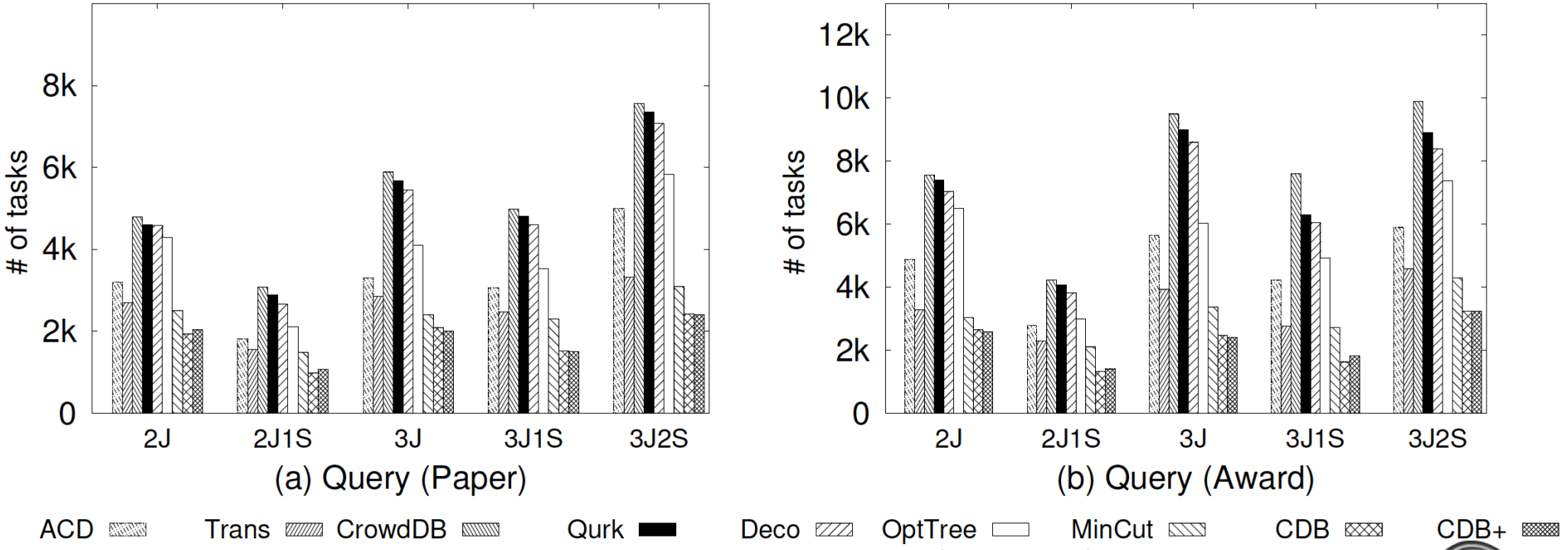


Figure 14: Varying Query (Real): # Tasks

Quality

Higher quality by about 5%

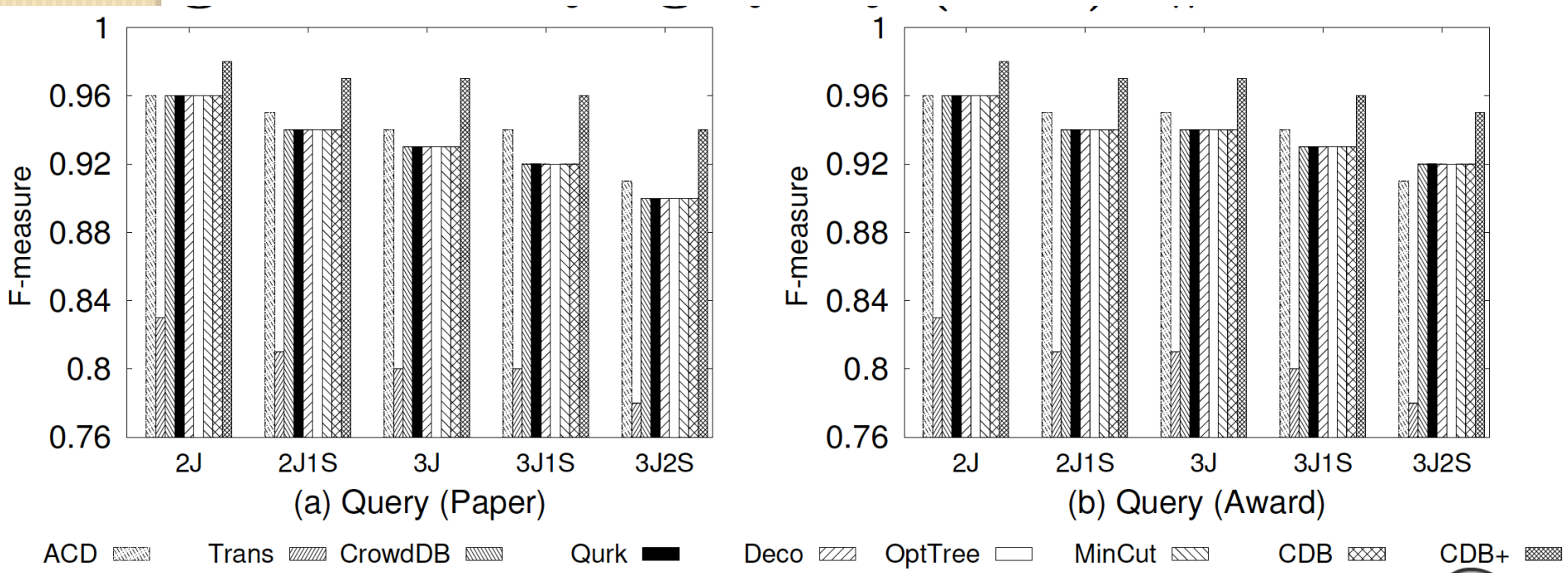


Figure 15: Varying Query (Real): Fmeasure

Latency

Lower Latency

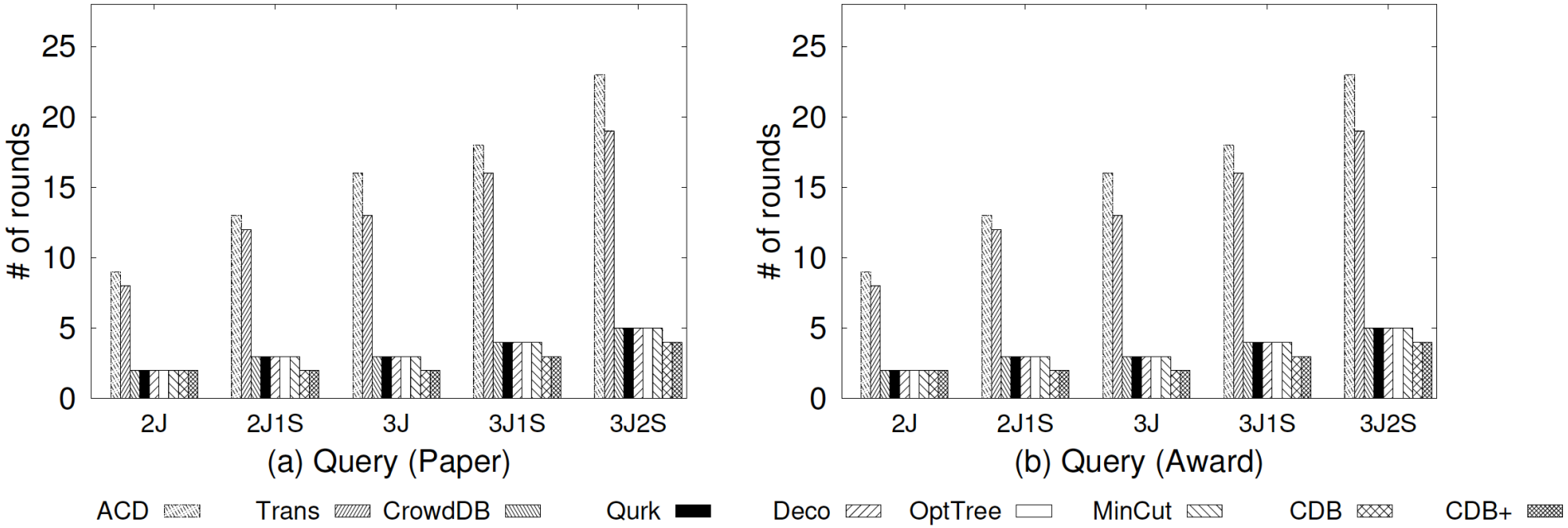


Figure 16: Varying Query (Real): # Rounds

Thank you!