

【SIGMOD2011勉強会】

Session 19: Ranking

担当：池田、大塚、三津石、安永（筑波大学）

セッション概要

1. On Pruning for Top-K Ranking in Uncertain Databases
 - ▶ 確率的データベースにおけるトップkランキング (PRF関数) の高速化
2. PathSim: Meta Path-Based Top-K Similarity Search in Heterogeneous Information Networks
 - ▶ 異種情報ネットワークにおける類似検索手法の提案
3. Optimizing and Parallelizing Ranked Enumeration
 - ▶ ランク付けの最適化問題解法アルゴリズム
Lawler-Murty's procedureの並列化
4. Efficient Rank Join with Aggregation Constraints
 - ▶ Rank Join (Top-k queries) での集約制約を用いた効率化

On Pruning for Top-k Ranking in Uncertain Databases

Chonghai Wang, Li Yan Yuan, Jia-Huai You, Osmar R
Zaiane (University of Alberta),
Jian Pei (Simon Fraser University)

研究の目的

生成ルール r

- ▶ **ひとことと言うと？**
確率的データベースにおけるトップkランキング手法であるPRFをPruningして高速化

	Time	Radar	Model	Plate No	Speed	Prob
t_1	11:45	L1	Honda	X-123	120	1.0
t_2	11:50	L2	Toyota	Y-245	130	0.7
t_3	11:35	L3	Toyota	Y-245	95	0.3
t_4	12:10	L4	Mazda	W-541	90	0.4
t_5	12:25	L5	Mazda	W-541	110	0.6
t_6	12:15	L6	Chevy	L-105	105	0.5
t_7	12:20	L7	Chevy	L-105	85	0.4

※ Figure 1: A sample uncertain database

※論文中のFigure1から引用

- ▶ **PRF(Parameterized Ranking Functions(VLDB '09))**
トップkランキングの様々な手法を, パラメータを調整することで近似的にシミュレートするランキング関数

$$PRF^\omega : \Upsilon(t) = \sum_{W \in PW(t)} \omega(t, \beta w(t)) \times \Pr(W)$$

PW(t) : tを含む全ての可能世界のセット
 $\beta w(t)$: 可能世界Wにおけるtの位置
 $\omega(t, i)$: 重み関数

キーアイデア

- ▶ 確率的データベース T が与えられた時に, q 個のタプルのセット($Q = \{t_1, \dots, t_q\}$)と, それらに関連づけられた生成ルール r ($R = \{r_1, \dots, r_l\}$)を考える.
任意の $t \in Q$ に対し, それらのUpper Boundを見つけることが目的.
そのため次の式を満たすような実数 c_i を求める

$$\sum_{i=1}^q c_i \Upsilon(t_i) \geq 0$$

- ▶ $c < 0$ の場合, 次のように変形可能

$$\Upsilon(t) \leq \sum_{t_i \in Q, t_i \neq t} -\frac{c_i}{c} \Upsilon(t_i)$$

- ▶ この時, t がUpper Boundにあたる

評価実験

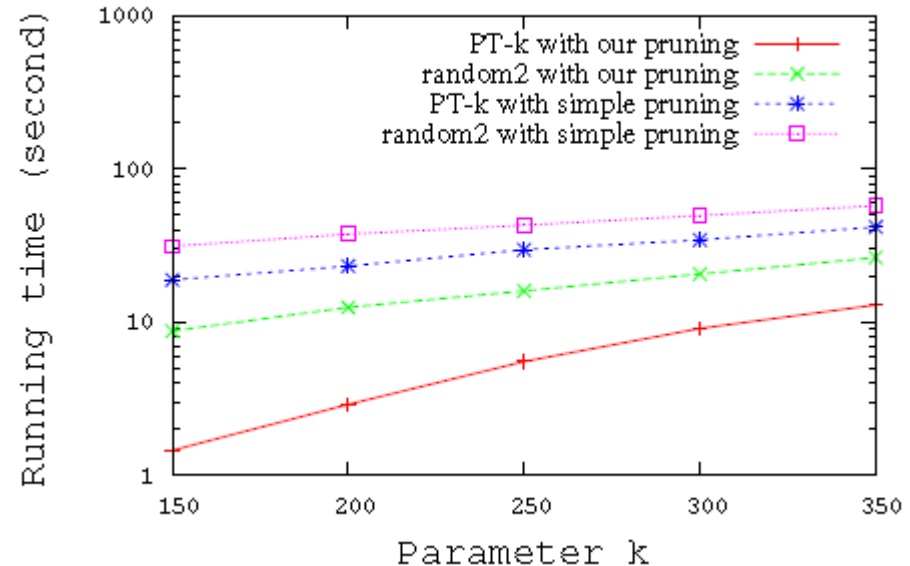
▶ 実験内容

2つのランキング方法
(random2, PT-k)に対して
kなどを変えて計算する
タプル数, 実行時間などの
比較を行う

▶ 結果

特にPT-k関数に対し素晴らしい高速化を実現

※(c)Running time and k



※論文中 (<http://www.cs.sfu.ca/~jpei/publications/TopKRankingVLDB11.pdf>) のFigure6から引用

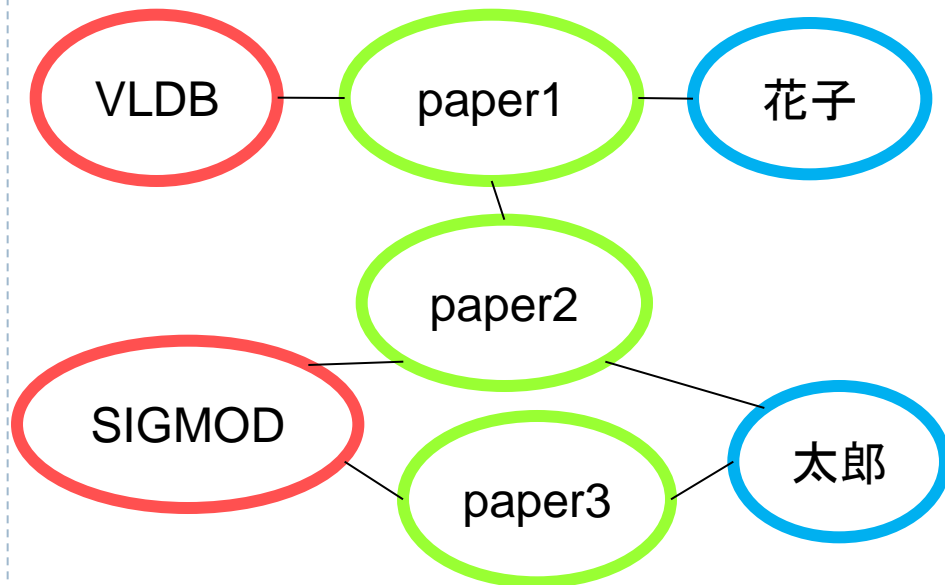
[2]PathSim: Meta Path-Based Top-K Similarity Search in Heterogeneous Information Networks

Yizhou Sun (UIUC), Jiawei Han (UIUC), Xifeng Yan (UCSB), Philip Yu (UIC), Tianyi Wu (Microsoft)

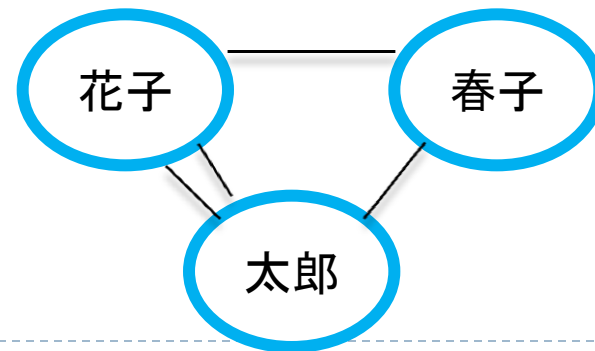
研究の目的

- ▶ ひとことと言うと？
 - ▶ 異種情報ネットワークにおける類似検索手法の提案
- ▶ 類似検索
 - ▶ この著者と領域・評判が似ている著者は？
- ▶ 同種ネットワークにおける類似検索
 - ▶ オブジェクト間のパスの多さ
 - ▶ オブジェクト間の近さ
 - ▶ 異なるタイプのオブジェクトは区別できない
- ▶ 提案する類似検索
 - ▶ パスのセマンティクスを考慮
 - ▶ top-k 検索の処理の効率化

異種情報ネットワークの例: DBLPネットワーク



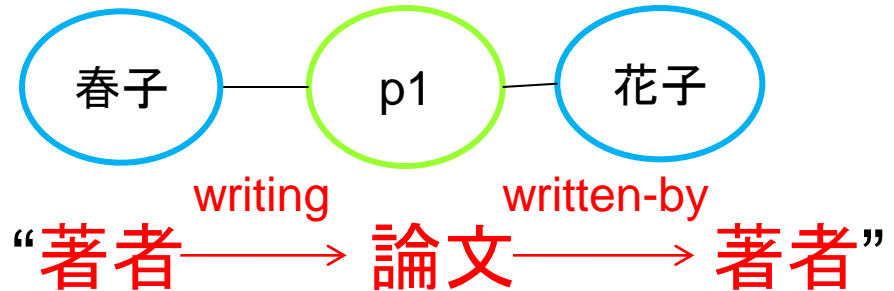
同種ネットワークの例: コミュニティネットワーク



キーアイデア

▶ Meta Path

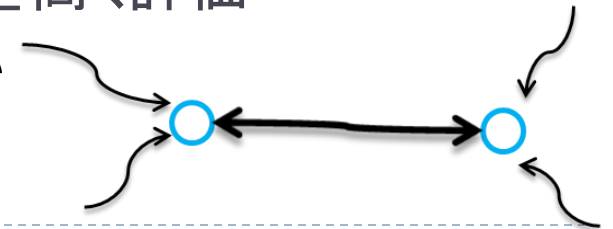
- ▶ 2つのオブジェクト間のパスをオブジェクトと関係の**種類の並び**で記述
- ▶ 共著者の関係のMeta Path



- ▶ オブジェクト間の新しい関係(⇒top-k 検索処理の高速化に使用)
- ▶ 提案する類似検索手法でMeta Path を使う

▶ PathSim

- ▶ 以下の特徴をもつオブジェクト間の類似性を高く評価
 - ▶ 指定した Meta Path に属するパスの数が多い
 - ▶ 見た目のバランスが似ている



評価実験

▶ 対象データ: DBLPネットワーク, Flickrネットワーク

- A) 提案手法と既存手法それぞれの検索結果の比較
入力: オブジェクト, Meta Path
出力: 類似度の高い順
→ 見比べると提案手法が良

- B) Meta Pathの長さの影響
→ $MP > (MP)^2 > (MP)^3$
- C) Meta Pathがセマンティクスを表せているか
→ 下図

MP: 画像-タグ-画像[2]



(a) top-1

(b) top-2

(c) top-3



(d) top-4

(e) top-5

(f) top-6

MP: 画-タグ-画-グループ-画-タグ-画[2]



(a) top-1

(b) top-2

(c) top-3



(d) top-4

(e) top-5

(f) top-6

Optimizing and Parallelizing Ranked Enumeration

Konstantin Golenberg (The Hebrew University), Benny Kimelfeld (IBM Research - Almaden), Yehoshua Sagiv (Hebrew University, Jerusalem)

Optimizing and Parallelizing Ranked Enumeration

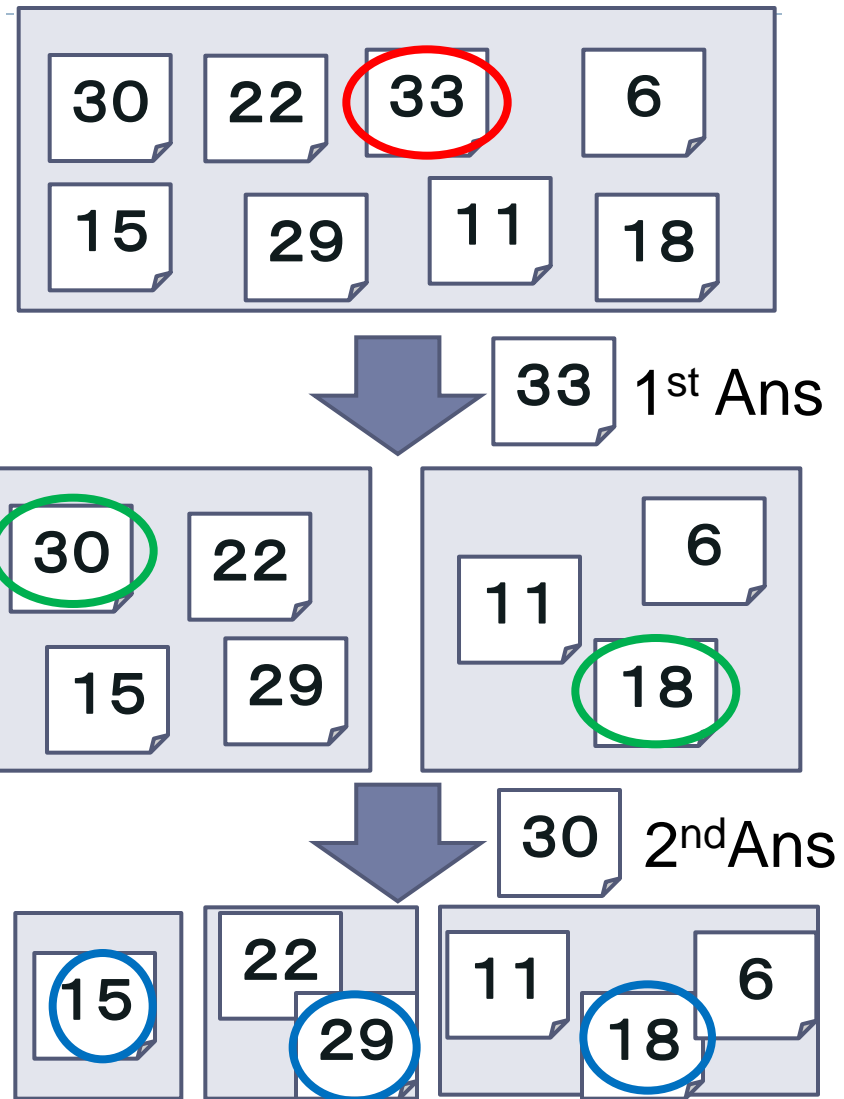
研究の目的

▶ ひとことでは言うとは？

- ▶ ランク付けの最適化問題
解法アルゴリズム
Lawler-Murty's procedure
の並列化

▶ Lawler-Murty's procedure

- ▶ 問題に対するAnswer列挙
 - ▶ Keywordサーチなど
- ▶ Answerの結果をソート
上位k個並べる → コスト大
- ▶ 上位1個のみ → 最適化問題
→データを制約により分けし
最適化問題を解く

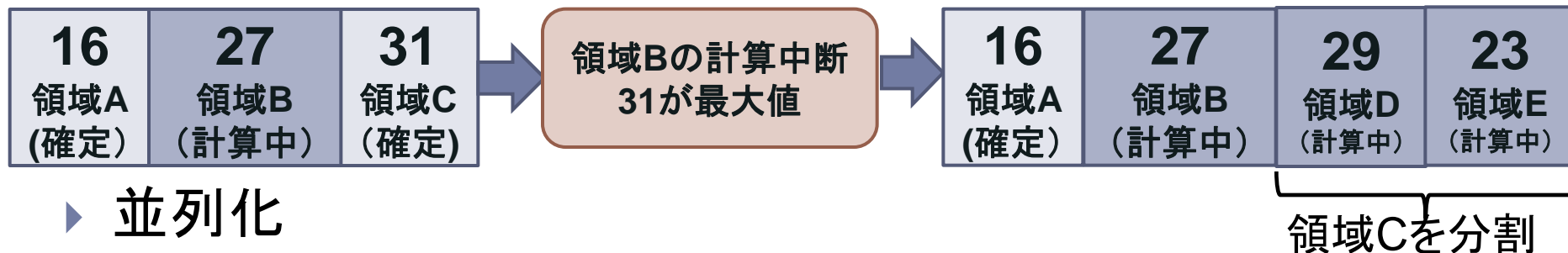


Optimizing and Parallelizing Ranked Enumeration

キーアイデア

▶ Freezing

- ▶ 区分けした各領域での最大値計算は領域ごとに処理時間が異なる → すべての領域で計算が終了するまで待機
- ▶ 計算中の領域の現時点での最大値が、他の領域の最大値より小さい場合、計算を中断 (Freezing) する



▶ 並列化

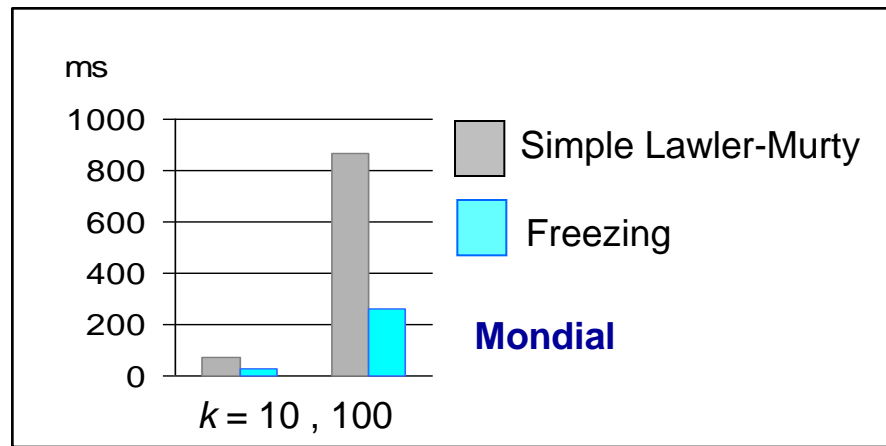
- ▶ 各領域の計算をスレッドに割り当てただけでは処理速度はほとんど向上しない → idle状態の発生
- ▶ Computed AnswerタスクとFrozenタスクにわけ、Frozenタスクでは、複数のスレッドで最大値計算を行なっておく。Computed Answerタスクで空きができた時点でFrozenタスクにある中で最もスコアの大きい物とComputed Answerタスクにあるスコアを比較

Optimizing and Parallelizing Ranked Enumeration

評価実験 (VLDB2011発表スライド Session9-3より)

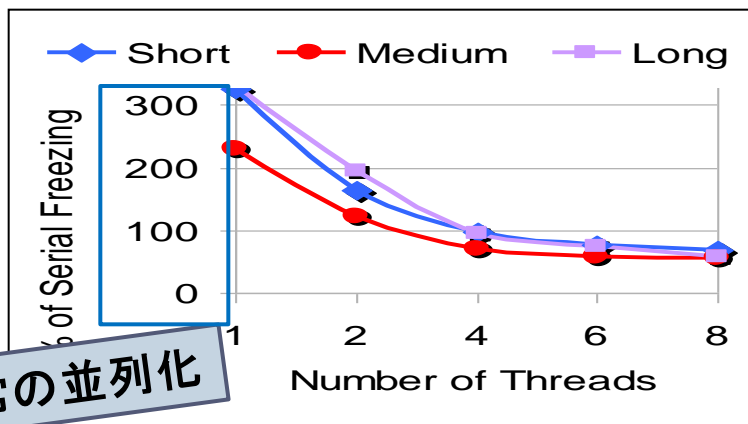
▶ Simple Lawler-Murty vs. Freezing

- ▶ グラフ探索結果の処理時間の比較
- ▶ Freezingにより **処理時間を約56%に削減**

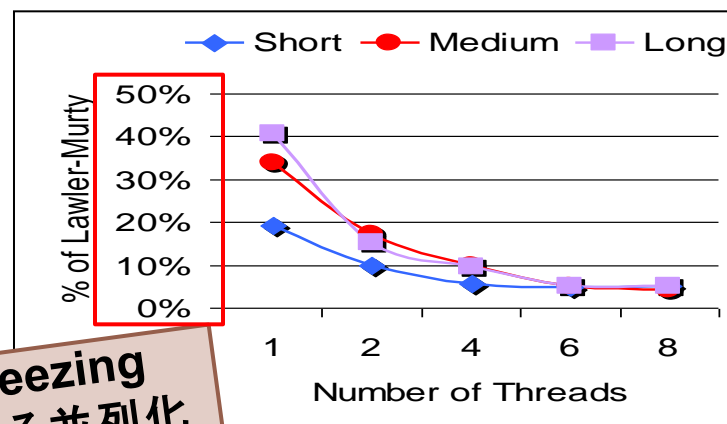


▶ SingleCore Freezing vs. MultiCore

- ▶ 通常の並列化では4core使用して1coreのFreezingと同程度のパフォーマンス
- ▶ Freezingを組み合わせた並列化では **8スレッドでFreezingの約5%の処理時間**



通常の並列化



Freezingによる並列化

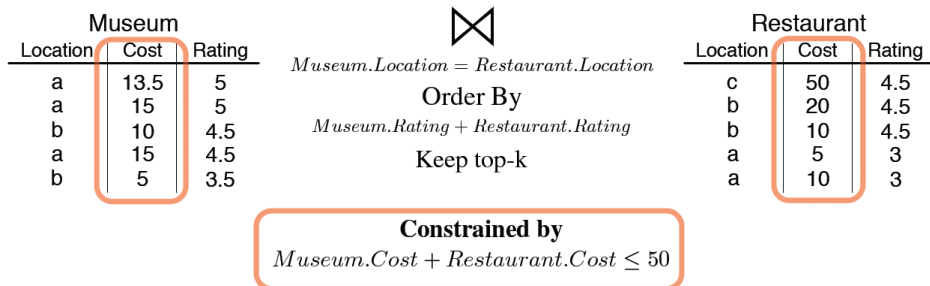
Efficient Rank Join with Aggregation Constraints

Min Xie (University of British Columbia), Laks Lakshmanan (University of British Columbia), Peter Wood (Birkbeck, University of London)

研究の目的

▶ ひとことでは言うとは？

Rank Join (Top-k queries) における
集約制約 (Aggregation Constraints) を
用いた効率的アルゴリズム
の提案



▶ Rank Joinのアプリケーション例

package recommendation

▶ 集約制約の例

$Museum.Cost + Restaurant.Cost \leq 20$

※集約とは、結合結果リレーションにおける各タプル内の値の「集約」

集約制約を用いた
Rank Joinの例

<http://www.vldb.org/2011/files/slides/research19/rSession19-4.pdf> から引用

キーアイデア

- ▶ Rank Joinを実行する際には、多くの状況で集約制約は自然に (naturally) でてくるもの
 - 集約制約を用いRank Joinの実行時間を効率化できるとよい
 - 集約制約を用いた効率的アルゴリズムを提案

▶ 提案アルゴリズム

▶ 決定的アルゴリズム

(2つの枝刈り戦略)

1. SubS-Pruning
2. Adaptive SubS-Pruning

▶ 確率的アルゴリズム

Museum			Restaurant			Constraint
Location	Cost	Rating	Location	Cost	Rating	$SUM(Cost, true) \leq 20$
t_1 : a	13.5	5	t_6 : c	50	4.5	Tuple Pruned $\{t_4\}$
t_2 : a	15	5	t_7 : b	20	4.5	
t_3 : b	10	4.5	t_8 : b	10	4.5	
t_4 : a	15	4.5	t_9 : a	5	3	
t_5 : b	5	3.5	t_{10} : a	10	3	

Figure 3: Tuple pruning using aggregation constraints.

Museum			Restaurant			Constraint
Location	Cost	Rating	Location	Cost	Rating	$SUM(Cost, true) \leq 20$
t_1 : a	13.5	5	t_6 : c	50	4.5	Top-1 result $\{t_3, t_8\}$ Tuple Pruned $\{t_2, t_4\}$
t_2 : a	15	5	t_7 : b	20	4.5	
t_3 : b	10	4.5	t_8 : b	10	4.5	
t_4 : a	15	4.5	t_9 : a	5	3	
t_5 : b	5	3.5	t_{10} : a	10	3	

Figure 4: Adaptive subsumption-based pruning.

評価実験

▶ 実験内容

4つのアルゴリズムを実装し、
実行速度と枝刈りを比較

(a) Post Filtering (ナイーブ)

(b) SubS-Pruning

(c) Adaptive SubS-Pruning

(d) Probabilistic

確率的アルゴリズムの結果の
質を評価

▶ 結果

▶ 提案アルゴリズムはいずれも
既存手法を上回る性能

▶ 確率的アルゴリズムは高い

▶ 18 質の結果を返す

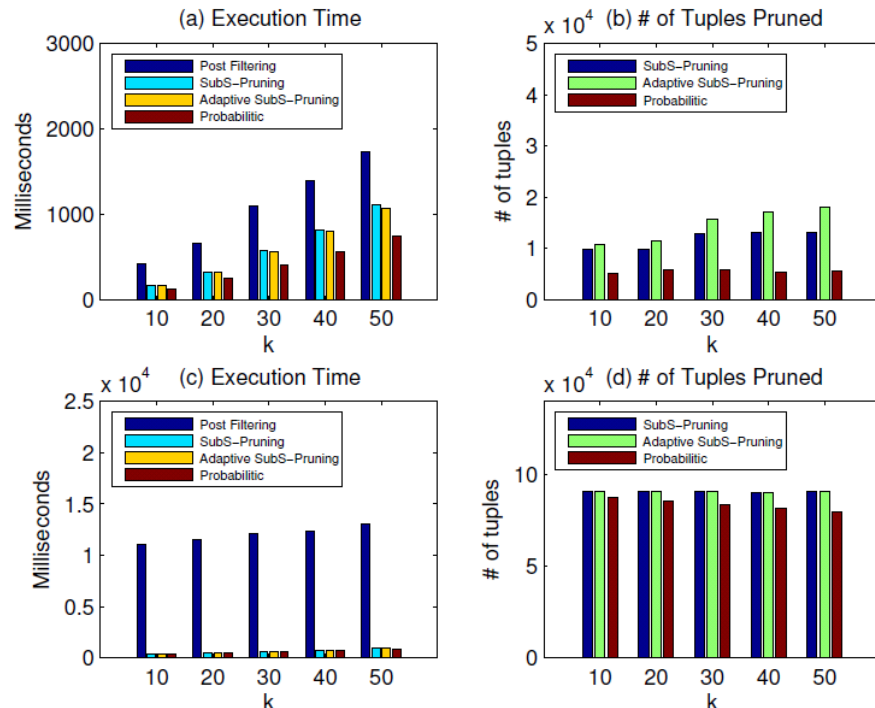


Figure 6: Uniform dataset: (a), (b) $SUM(A, true) \geq \lambda$, selectivity 10^{-5} ; (c), (d) $MIN(A, true) \leq \lambda$, selectivity 10^{-5} .

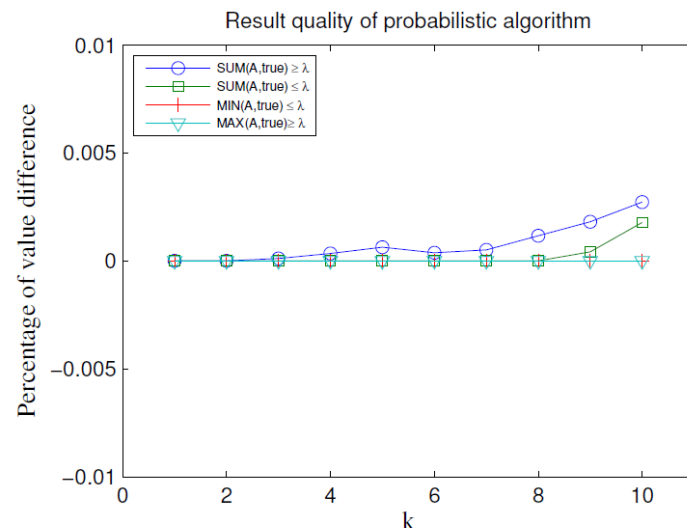


Figure 9: Quality of the probabilistic algorithm.