General Space-Time Tradeoffs via Relational Queries

Shaleen Deep[1], Xiao Hu[2], and Paris Koutris[3]
[1]Microsoft
[2]University of Waterloo
[3]University of Wisconsin-Madison

WADS, 2023

INTRODUCTION

Developing faster data structures for set intersection, reachability oracles, document indexing etc. is an important problem

A *preprocessing phase* computes a space efficient data structure (space S) and the *answering phases* uses the data structure to complete the task as fast as possible (time T)

Fundamental question: What is the tradeoff between S and T?

INTRODUCTION

2-set Disjointness problem: Given m sets $C_1, \ldots, C_m$ of total size N, create a data structure to answer whether C_i and C_j intersect or not for any i, j in [m]

Best known tradeoff: $S \times T^2 = N^2$

k-reachability problem: Given a directed graph G=(V,E), decide whether there is a path of length k between between any two vertices u and v

Best known tradeoff: $S \times T^{2/(k-1)} = N^2$

These problems have been studied in isolation and the results are not generalizable due to lack of a comprehensive framework

OUR CONTRIBUTION

A Comprehensive Framework. We show that several widely-studied data structure problems can be captured as *queries over a database*, allowing us to use the formalism of Conjunctive Queries and results from the Database community

Improved Algorithms. We explicitly improve the best-known space-time tradeoff for the k-reachability problem for any $k \geq 4$, the first non-trivial improvement for the problem

Conditional Lower Bounds. We show conditional lower bounds for k-reachability problem

TALK OUTLINE

Problem Setting and Main Result

Using Query Decompositions

Improved Tradeoffs for k-reachability

Future Work and Conclusion

TALK OUTLINE

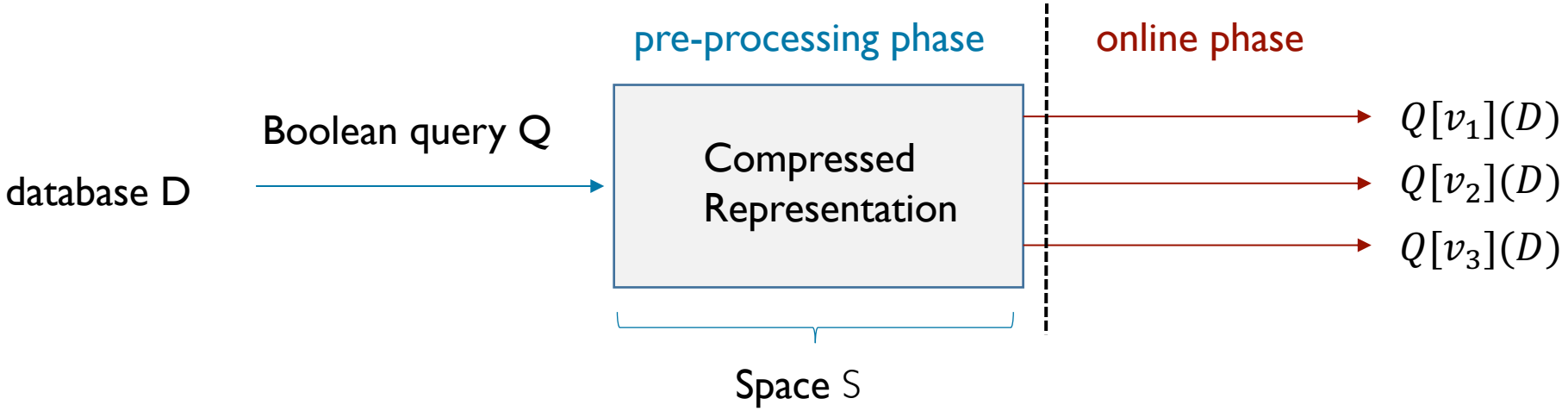
Problem Setting and Main Result

Using Query Decompositions

Improved Tradeoffs for k-reachability

Future Work and Conclusion

PROBLEM SETTING



Goal – Construct a space-efficient representation to answer the query

Parameters

- Space requirement S
- Query answering time T

ADORNED VIEWS

We consider the class of ***Conjunctive Queries***

$$V(x, z) = R(x, y), R(y, z)$$

An adorned view describes the access pattern where some variables are ***bound*** (denoted *b*)

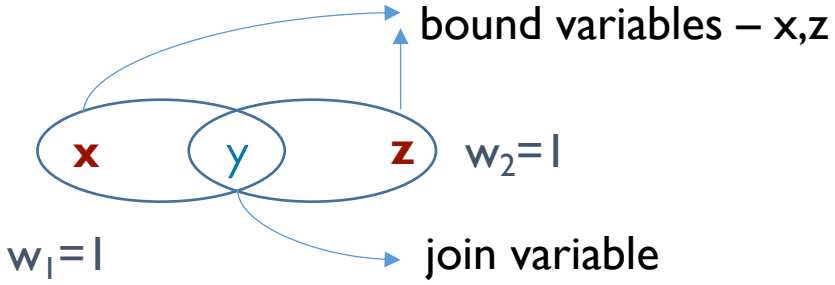
$$V^{bb}(x, z) = R(x, y), R(y, z)$$

In this paper, we consider Boolean adorned queries

EDGE COVERS AND AGM BOUND

- It is convenient to view adorned query as a hypergraph

$$V(x,z) = R(x,y), R(y,z)$$



$$\alpha(S = \{y\}) = 2$$

$$\mathbf{u} = (1,1)$$
$$\text{AGM bound} = |D|^2$$

- **Fractional edge cover** – assign a weight to each hyperedge such that for each variable, the sum of weights of hyperedges including it is at least one

- **Slack** – given a fractional edge cover **u**, slack $\alpha(S)$ is the largest quantity such that **u**/$\alpha(S)$ is still a valid fractional cover of S

- **AGM Bound** – given a fractional edge cover **u**, the output size of the query is upper bounded by $\Pi_{F \in \mathcal{E}} |R_F|^{u_F}$

MAIN RESULT

Consider the Boolean adorned view Q with hypergraph H = (V, E). Let **u** be any fractional edge cover of V. Then, for any database D, there exists a data structure that can answer in time T and takes space

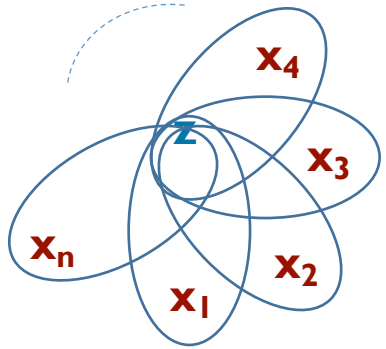
$$S = O(|D| + \Pi_{F \in \mathcal{E}} |R_F|^{u_F} / T^{\alpha})$$

Here, the slack is computed over all non-bound variables

Example: For 2-set intersection, we get $S = O(|D| + |D|^2/T^2)$ since each relation gets edge cover of one and the slack is two

APPLYING MAIN RESULT

For k-set intersection



$$\mathcal{V}_b = \{x_1, \ldots, x_n\}$$

$$\mathbf{u} = (1, \ldots, 1)$$

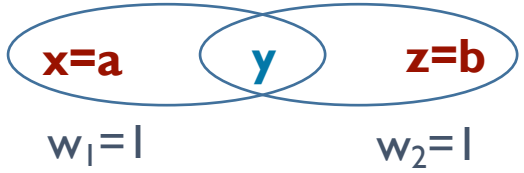
$$\alpha(\{z\}) = n$$

$$S = O(|D| + |D|^n / T^n)$$

DATA STRUCTURE

Key Observation – worst-case optimal join algorithms provide a time guarantee

$$V^{bb}(x, z) = R(x, y), R(y, z)$$



X	Z	
a	c	d_a
a	e	
a	f	
...	...	
b	e	d_b
b	f	
b	g	

If $\min\{d_a, d_b\} \leq \sqrt{d_a d_b} \leq \tau$, then we can evaluate the join and get delay guarantee τ

Otherwise, for all (a,b) where $\sqrt{d_a d_b} > \tau$, we will store some information

- Store a bit to tell us whether the answer is true or false

TALK OUTLINE

Problem Setting and Main Result

Using Query Decompositions

Improved Tradeoffs for k-reachability

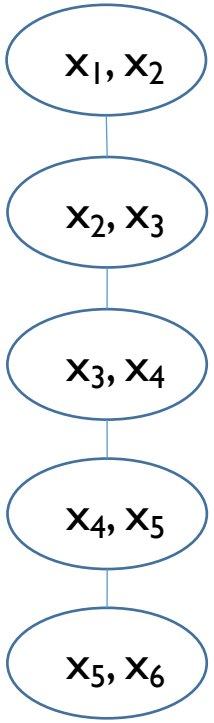
Future Work and Conclusion

QUERY DECOMPOSITION

$$Q^{bbb}(x_1, x_5, x_6) = R(x_1, x_2), \ldots, R(x_6, x_7)$$

Given a hypergraph H = (V, E), query decomposition is a tree

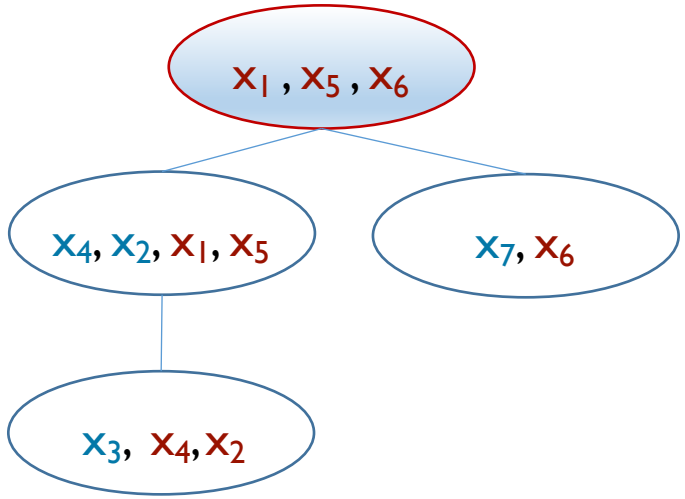
- each bag in the tree is a subset of V
- each edge in E is contained in some bag
- each variable in V is connected in the decomposition



C-connex is query decomposition (T, A)

- T is a tree decomposition
- A is a connected subset of nodes such that their bags contain exactly C

$$C = \mathcal{V}_b = \{x_1, x_5, x_6\}$$



TALK OUTLINE

Problem Setting and Main Result

Using Query Decompositions

Improved Tradeoffs for k-reachability

Future Work and Conclusion

IMPROVED TRADEOFFS

Consider the k-reachability problem expressed as a CQ

$$Q^{bb}(x_1, x_{k+1}) = R(x_1, x_2), \ldots, R(x_k, x_{k+1})$$

Goldstein et al. [WADS'17] showed an algorithm that can achieve $S \times T^{2/(k-1)} = N^2$

Key Idea: Fix a degree threshold d. The number of x_i values larger than d is at most N/d. For any x_1, x_{k+1} where both values are heavy, precompute the results (takes $(N/d)^2$ space). If (say) x_1 is light, recursively compute the data structure for subquery

$$Q^{bb}(x_2, x_{k+1}) = R(x_2, x_3), \ldots, R(x_k, x_{k+1})$$

IMPROVED TRADEOFFS

Our idea: $Q^{bb}(x_1, x_5) = R(x_1, x_2), R(x_2, x_3), R(x_3, x_4), R(x_4, x_5)$

Fix degree threshold d and construct a list $L(x_3)$ that contains all heavy x_3 values

Store the following two views (both takes N^2/d space)

$$V_1(x_1, x_3) = R(x_1, x_2), R(x_2, x_3), L(x_3)$$
$$V_2(x_3, x_5) = L(x_3), R(x_3, x_4), R(x_4, x_5)$$

For all light x_3, preprocess $Q^{bb}(x_1, x_5) = R(x_1, x_2), R'(x_2, x_4), R(x_4, x_5)$ using prior work. R' is the join of $R(x_2, x_3)$ and $R(x_3, x_4)$ for all light x_3 and takes space N.d

Thus, $S = O(N^2/d + N.d)$

IMPROVED TRADEOFFS

Our idea: $Q^{bb}(x_1, x_5) = R(x_1, x_2), R(x_2, x_3), R(x_3, x_4), R(x_4, x_5)$

(1) Given an (x_1, x_5) instantiation, check if (x_1, x_5) connects through $L(x_3)$ using the two views (takes time O(N/d))

$$V_1(x_1, x_3) = R(x_1, x_2), R(x_2, x_3), L(x_3)$$
$$V_2(x_3, x_5) = L(x_3), R(x_3, x_4), R(x_4, x_5)$$

(2) Answer $\underbrace{Q^{bb}(x_1, x_5) = R(x_1, x_2), R'(x_2, x_4), R(x_4, x_5)}$

3-path reachability – SxT = $O(N^2)$ tradeoff. For S=N.d, T=N/d

Final tradeoff: S = $O(N^2/d + N.d)$ and T = O(N/d), an improvement for $d \geq \sqrt{N}$

FUTURE WORK AND CONCLUSION

In this work we show how several popular problems can be cast as Database querying problems, allowing us to use results on query evaluation

We explicitly improved the tradeoff for k-reachability problem, the first non-trivial improvement for the problem

Future work: Understand how DB query evaluation literature can help build better approximate distance oracles.