



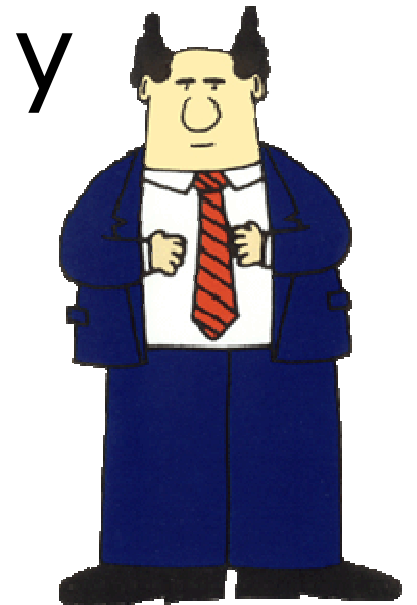
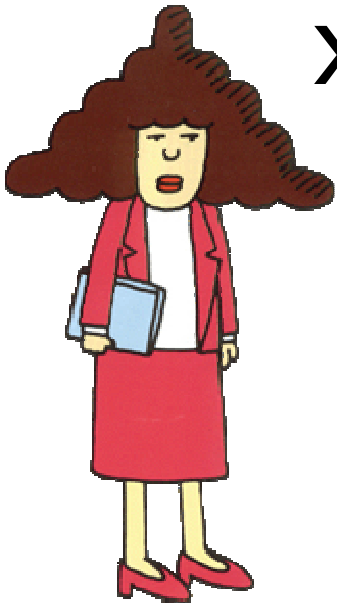
Private Approximation of Search Problems


Amos Beimel

Based on Joint works with
Paz Carmi, Renen Hallak, Kobbi Nissim,
and [Enav Weinreb](#)

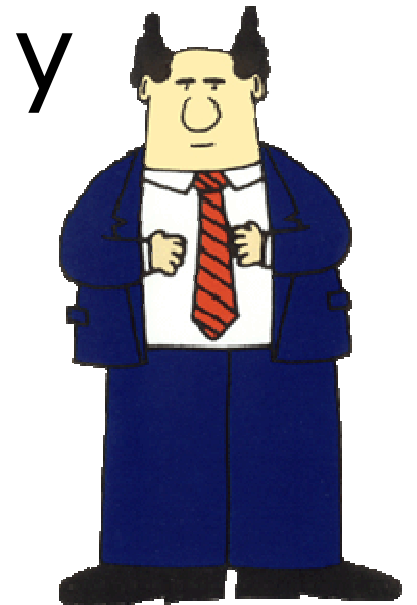
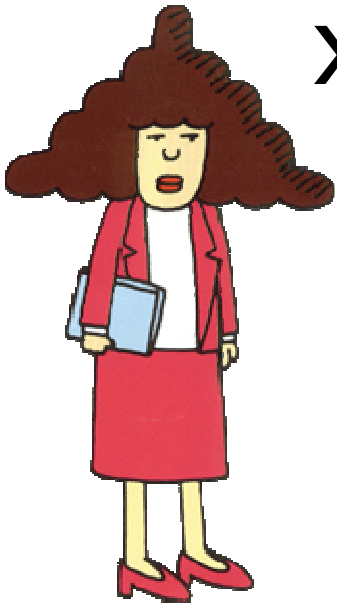



Let's compute $f(x, y)$!



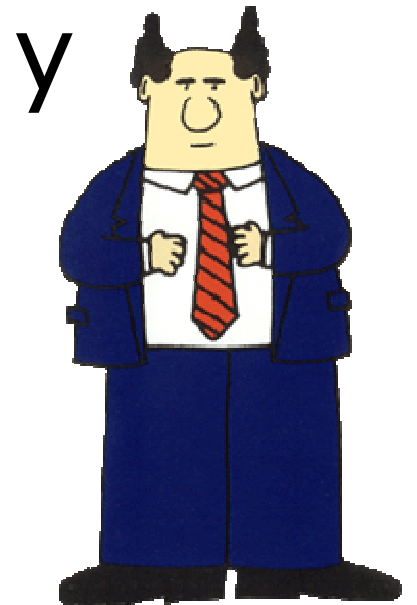


No! You will learn too much information on my input!



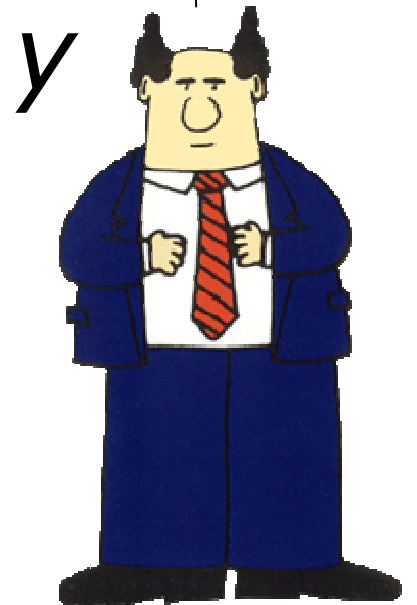



Haven't you heard of
secure function evaluation?



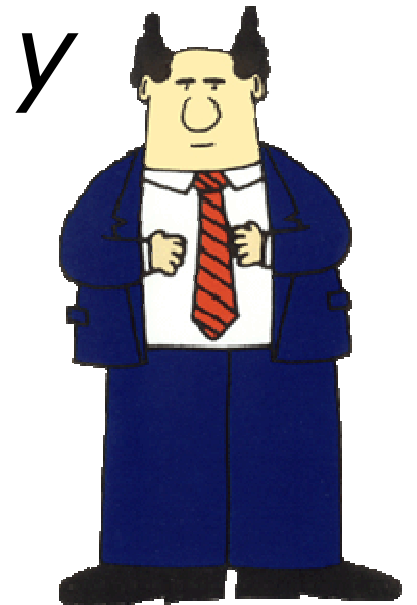


Sure I've heard of it...But
for f it will be inefficient





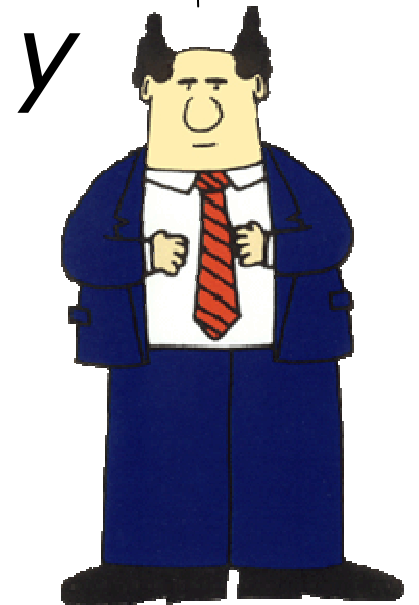
That's not a problem! We
can approximate f by f^*
and do SFE on f^* !





Hmmmm....

I don't know...



What can go wrong?

Example:

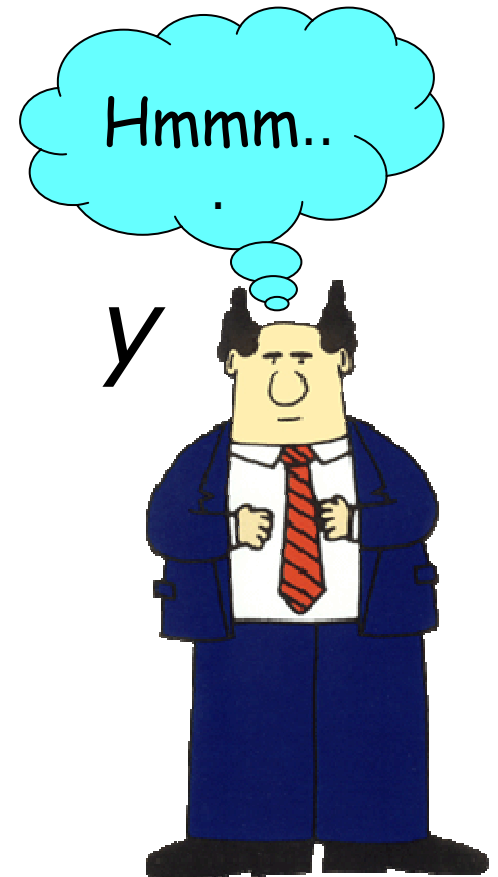
$f^*(x, y)$ reveals Bob's input.

$$f(134, 285) = 64$$

$$f^*(134, 285) = 64.285$$

$$f(847, 121) = 26$$

$$f^*(847, 121) = 26.121$$



Talk Overview



- i Background and Previous Work
- i Definitions for Search Problems
- i Impossibility Result for Vertex Cover
- i Algorithms that Leak (Little) Information
 - 1 Positive Result for MAX-3SAT
- i Problems in P
- i Conclusions and Open Problems



Private Approximation

[FeigenbaumIshaiMalkinNissimStraussWright01]

f^* is a **private approximation** for f :

- f^* is an approximation of f .
- $f^*(x)$ gives no more information about x than $f(x)$.

Privacy definitions:

- If $f(x)=f(x')$ then $f^*(x)$ and $f^*(x')$ should be **indistinguishable**.

Positive results [FIMNSW]

- i Hamming distance:
 - └ Private approximation in communication $O(\sqrt{n})$.
 - └ Improved to $\text{polylog}(n)$ [IndykWoodruff06]
- i Permanent:
 - └ Private approximation in polynomial time.





PA of NP-Hard Functions

[HaleviKrauthgamerKushilevitzNissim01]

Vertex Cover

Input: undirected graph $G = \langle E, V \rangle$.

A set $C \subseteq V$ is a **vertex cover** of G if
for every $\langle u, v \rangle \in E$, $u \in C$ or $v \in C$.

Functional:

Return **size** of minimum vertex cover.

* We'll discuss **search** version later.

Abstract Client-Server Model

Computes
Vertex
Cover C



SERVER

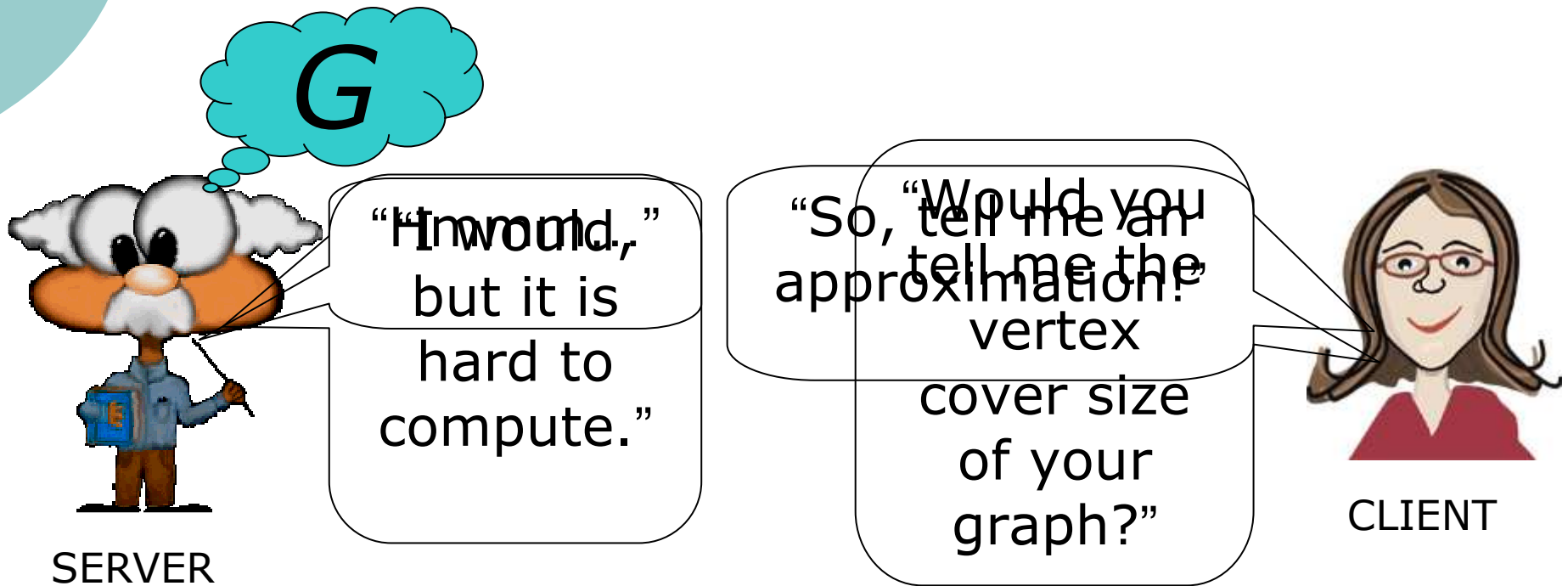
$|C|$



CLIENT

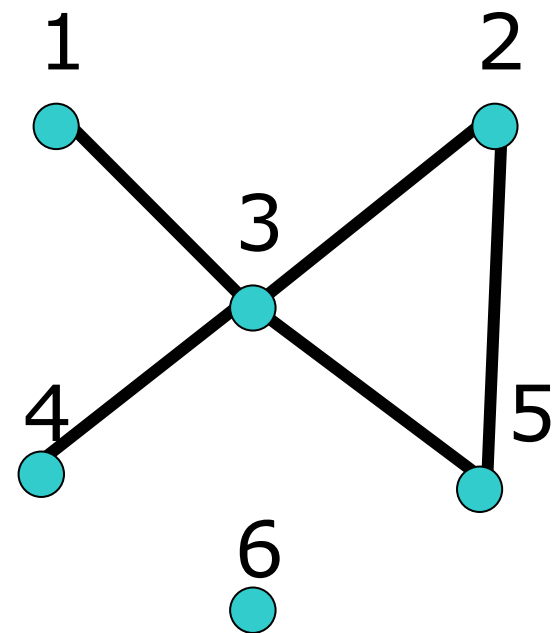
- i Impossibility in client-server model \Rightarrow Impossibility in multiparty.
- i Possibility in client-server model \Rightarrow Possibility in multiparty using SFE (Yao, Goldreich, Micali, Wigderson).

Client-Server Model



Maximal Matching Approximation

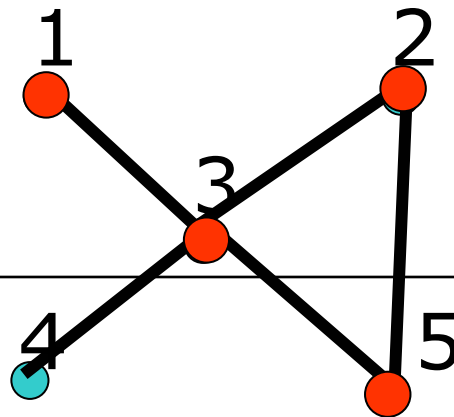
- i Find maximal matching.
- i Its vertices form a cover.
- i 2-approximation: solution size is at most 2 times the optimal solution.





VC

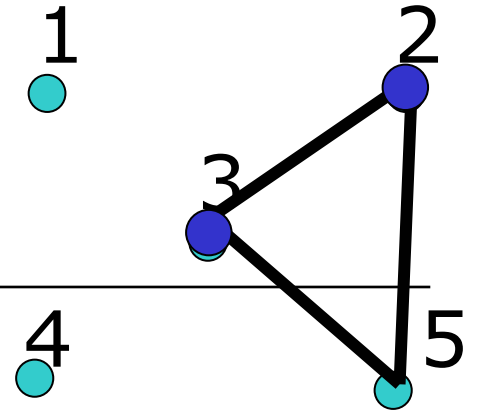
Matching



6

2

4



6

2

2



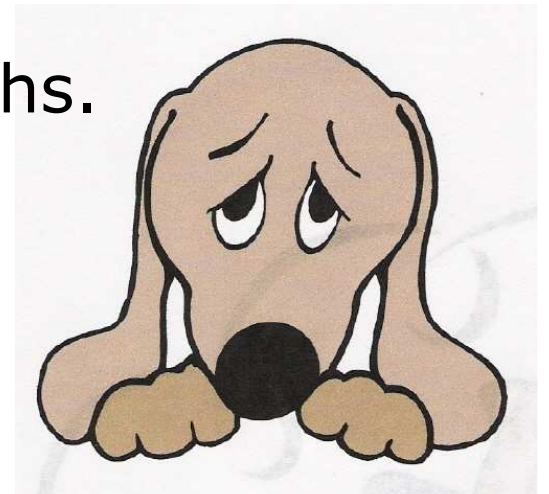
"Hmmm..."

"So, tell me an approximation!"



Impossibility results [HKKN]

- i If $\text{NP} \not\subseteq \text{BPP}$ **there is no** polynomial private $n^{1-\epsilon}$ -approximation algorithm for vertex cover size.
- i Impossibility results for other NP-complete functions:
 - 1 MAX-SAT
 - 1 Vertex cover in planer graphs.



Talk Overview



- i Background and Previous Work
- i **Definitions for Search Problems**
- i Impossibility Result for Vertex Cover
- i Algorithms that Leak (Little) Information
 - 1 Positive Result for MAX-3SAT
- i Problems in P
- i Conclusions and Open Problems



Search problems

- Function – one output for every input.
- Search – many solutions for one input.

Example: vertex cover

Return a vertex cover of the graph (a set of vertices).

- What is the right definition of privacy?
- What pairs of inputs should not be distinguished by the output?

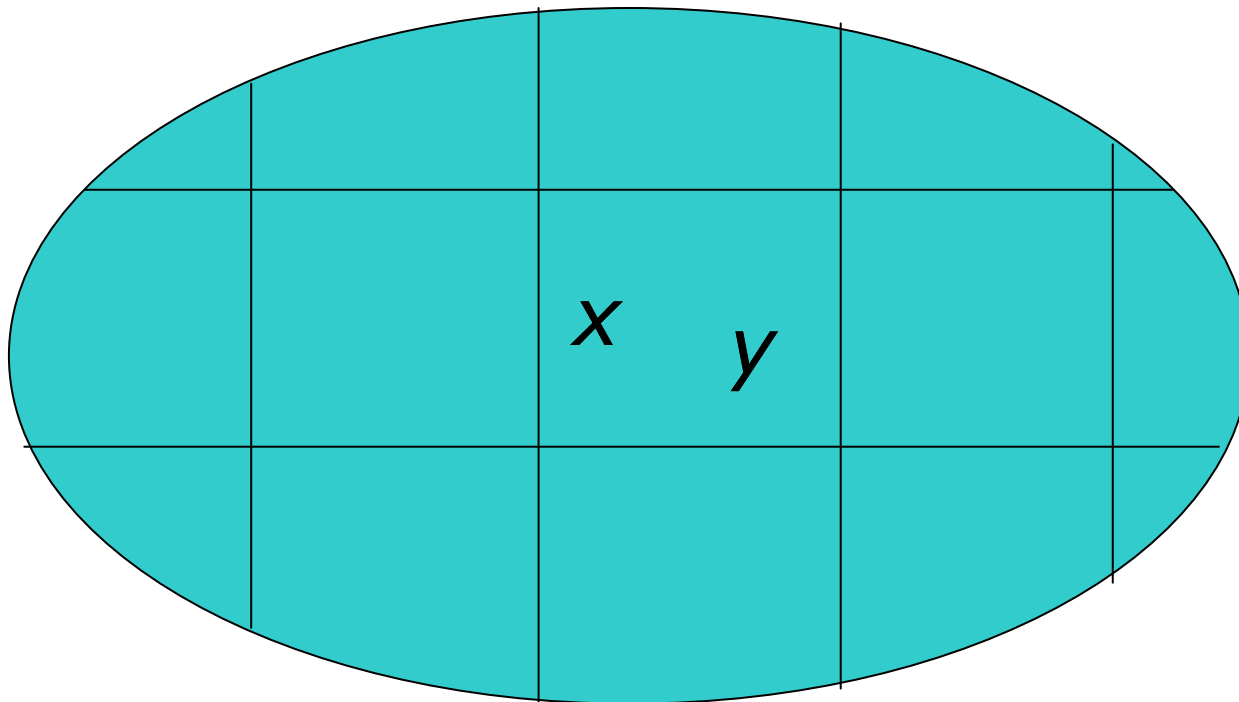


Step 1: Privacy w.r.t. a Relation

R – Equivalence relation over the inputs

\mathbf{A} – Probabilistic algorithm

\mathbf{A} is **private with respect to** R if:



$$\mathbf{A}(x) \approx^{\epsilon} \mathbf{A}(y)$$



Step 2: Defining the Relation

Let P be a search problem.

Let $S(x)$ be the set of solutions for the input x .

We say that $x \approx_P y$ if x and y have the same set of solutions, that is,
 $S(x) = S(y)$.

Example – Vertex Cover (Search)

- $G_1 \approx_{VC} G_2$ if they have **the same set** of minimum vertex covers.

- A is a **private approximation** algorithm for vertex cover if:

- A is an approximation algorithm for vertex cover.

$$G_1 \approx_{VC} G_2 \iff A(G_1) \approx A(G_2)$$

- Can this be done efficiently?

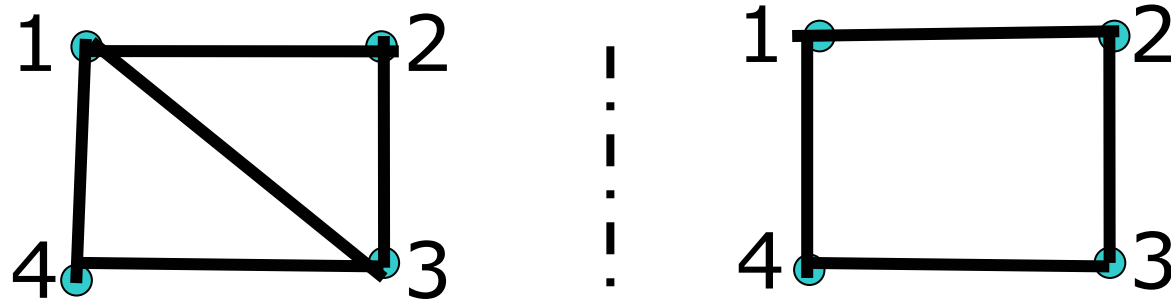
vertex cover sets:

$\{2,3\}$ and $\{3,5\}$



Search versus Functional

- i In non-private computation:
 - 1 Infeasibility of **functional** implies infeasibility of **search**.
- i Private computation:



- 1 **Functional** – equivalent (VC size = 2).
- 1 **Search** – not equivalent ($\{2,4\}$ is a VC only of the right graph).



Search versus Functional

Can we use the lower bounds techniques of [HKKN] for functional vertex cover?

No.

- ; [HKKN] relies on having few equivalence classes.
- ; In search - **Huge** number of equivalence classes.

Talk Overview



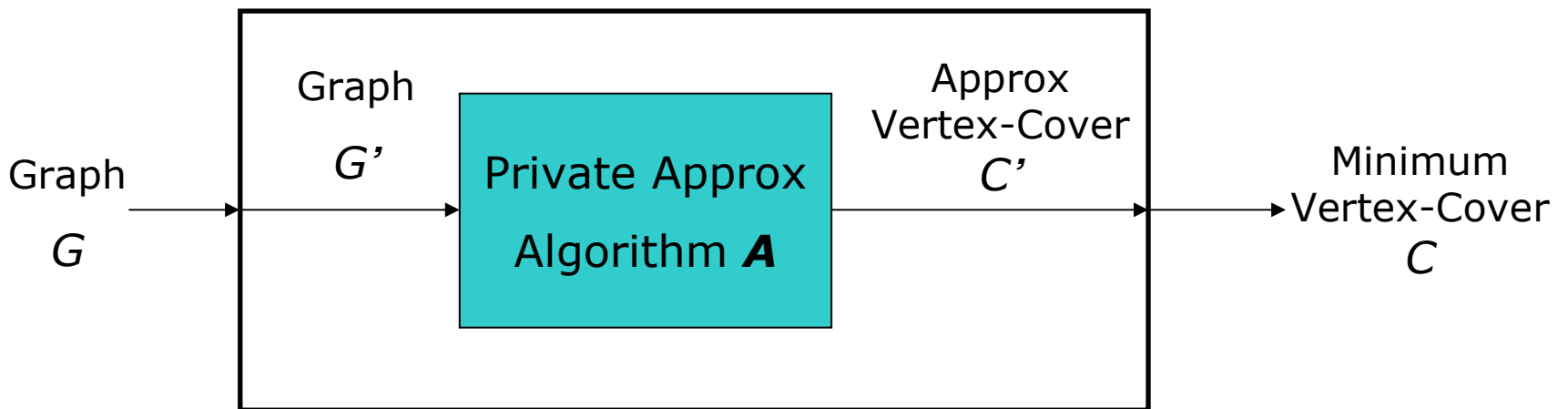
- i Background and Previous Work
- i Definitions for Search Problems
- i **Impossibility Result for Vertex Cover**
- i Algorithms that Leak (Little) Information
 - 1 Positive Result for MAX-3SAT
- i Problems in P
- i Conclusions and Open Problems



Vertex Cover - Impossibility Result

Thm 1: If $\text{RP} \neq \text{NP}$ there is no deterministic polynomial time private $n^{1-\epsilon}$ -approximation algorithm for vertex cover – search version.

Proof idea:



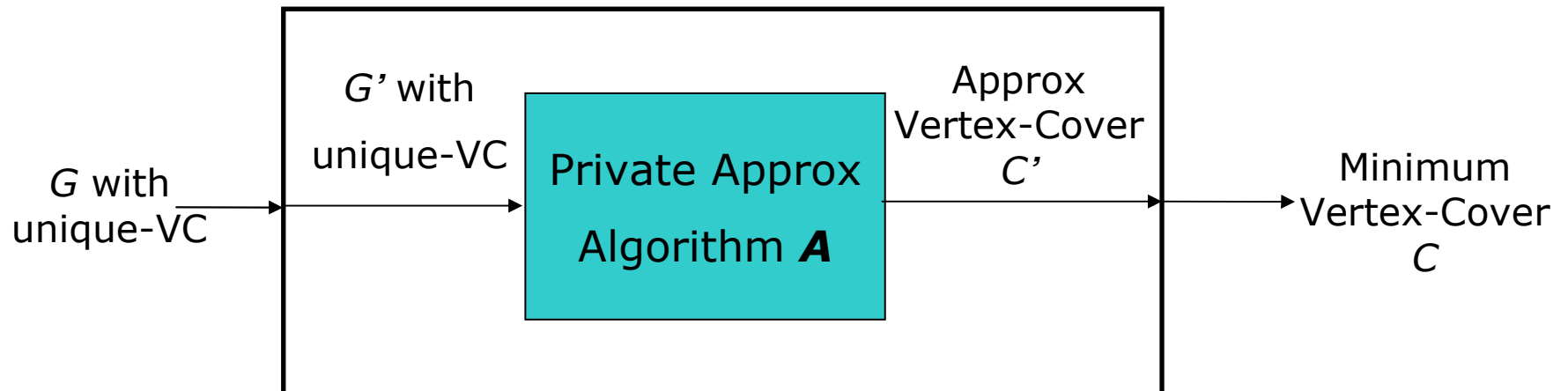
First Tool: Unique-Vertex-Cover

Input: A graph G

Promise: G has a unique minimum vertex cover

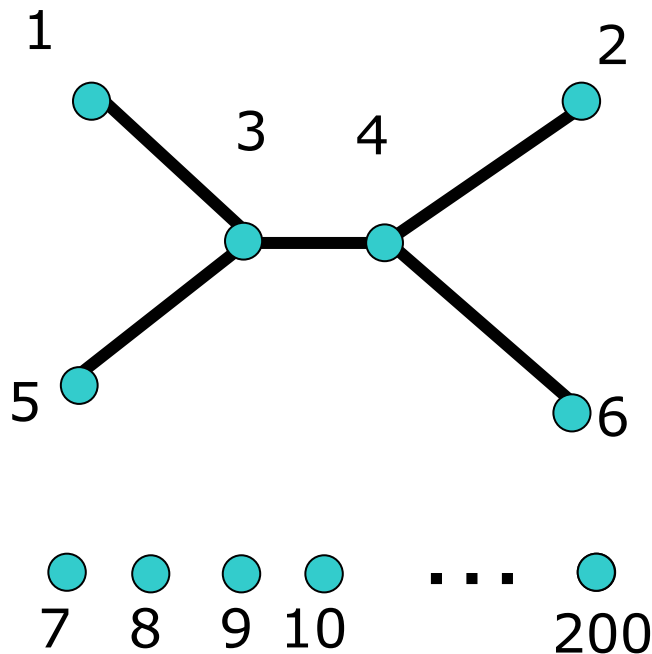
Output: The minimum vertex cover

Thm [ValiantVazirani86]: Solving Unique-Vertex-Cover is NP-hard.

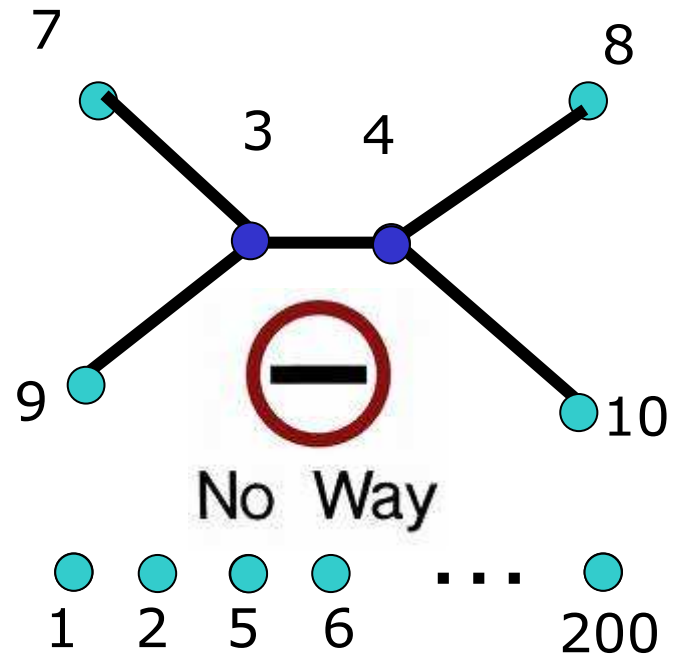


Second Tool: Adding Vertices

Claim 1: Private Approx Alg **A** must return VC + other vertices.



\approx_{VC}



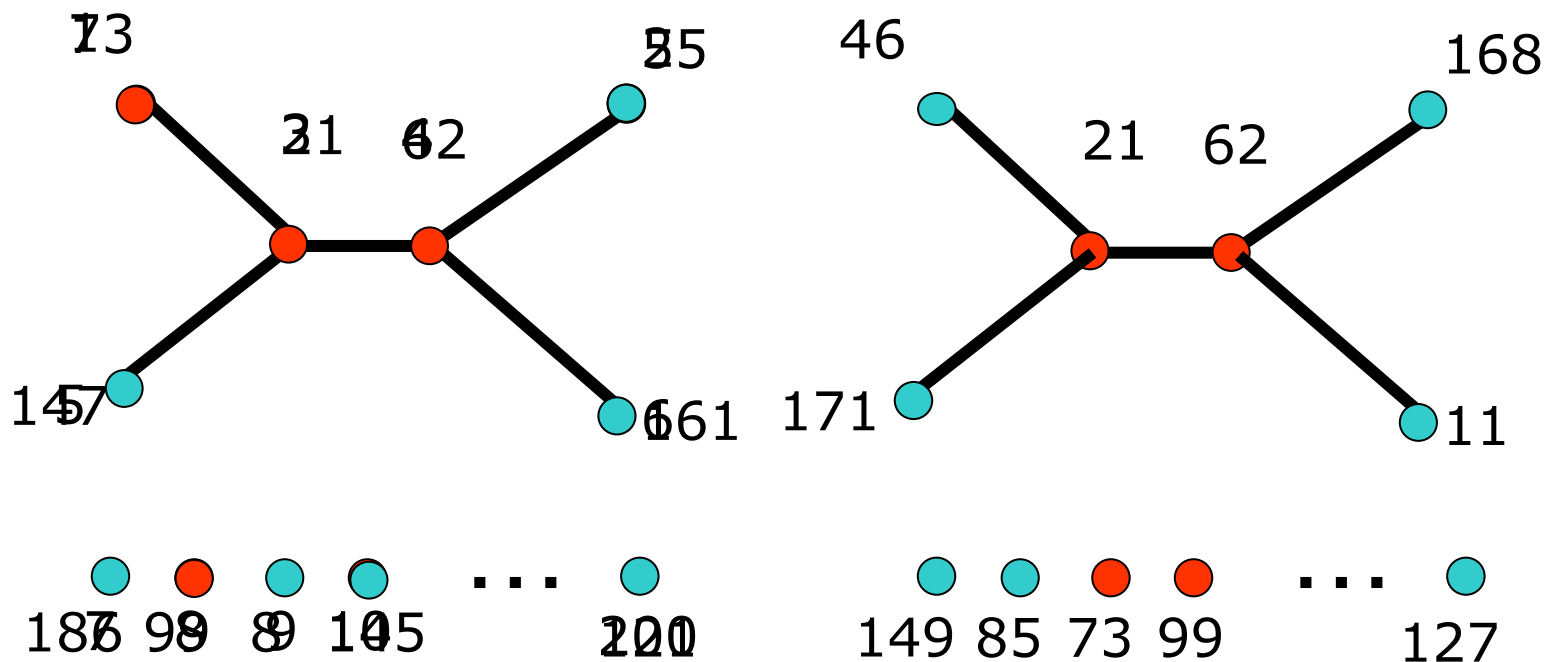
No Way



Claim 1: Private Approx Alg **A** must return VC + other vertices.

Third Tool: Random Renaming

Claim 2: With high probability, \mathcal{A} must return $WC + \text{isolated vertices}$.





Summary of Proof:

Thm 1: If $RP \neq NP$ there is no ~~deterministic~~ ^{randomized} polynomial time private $n^{1-\epsilon}$ -approx algorithm for vertex cover

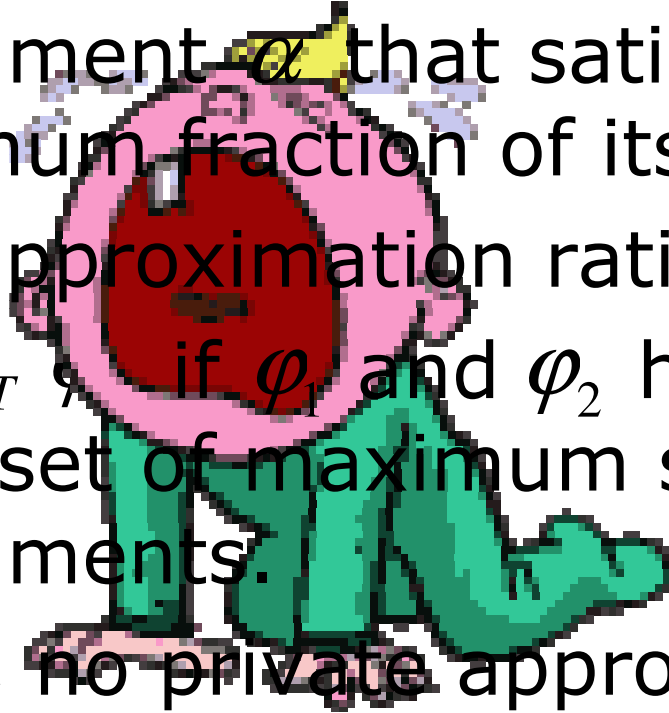
Proof:

- G – graph with unique VC
- Add isolated vertices to G
- Randomly permute names of vertices
- Execute $C' \leftarrow \mathbf{A}(G')$
- VC C of G – original vertices in C'

If $RP \neq NP$, then no such algorithm \Rightarrow NO **A**.

MAX-3SAT

- Given a 3CNF formula φ find an assignment α that satisfies the maximum fraction of its clauses.
- Best approximation ratio: $7/8$.
- $\varphi_1 \approx_{SAT} \varphi_2$ if φ_1 and φ_2 have the same set of maximum satisfying assignments.
- Again, no private approximation!



Talk Overview



- i Background and Previous Work
- i Definitions for Search Problems
- i Impossibility Result for Vertex Cover
- i **Algorithms that Leak (Little) Information**
 - 1 **Positive Result for MAX-3SAT**
- i Problems in P
- i Conclusions and Open Problems





Almost-Private Algorithms [HKKN]

- Let f be a **function**.
- f^* is an approximation for f that **leaks k bits**:
 - $f^*(x)$ can be simulated from $f(x)$ and another **k bits of advice**.

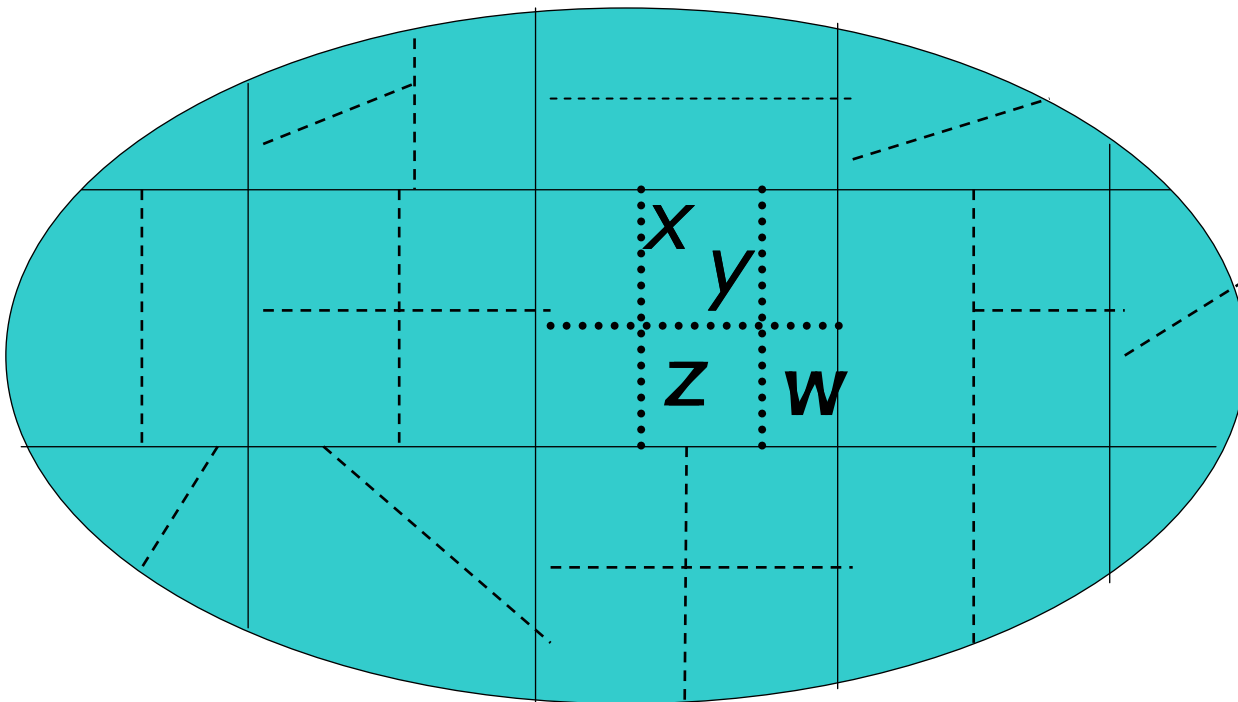
Example:

There is an efficient **4**-approximation of vertex cover size that leaks **1** bit.

Almost-Private Algorithms – Search

$$\mathbf{A}(\cdot) \approx \mathbf{A}(\cdot)$$

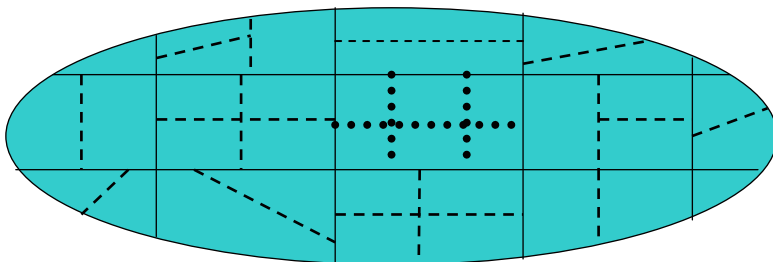
$$\mathbf{A}(\cdot) \stackrel{?}{=} \mathbf{A}(\cdot)$$



Almost-Private Algorithms

A is **leaks k bits** with respect to R if there exists R' such that:

1. $R' \subseteq R$.
2. Every equivalence class of R is a union of at most 2^k equivalence classes of R' .
3. **A** is private with respect to R' .





Search versus Functional

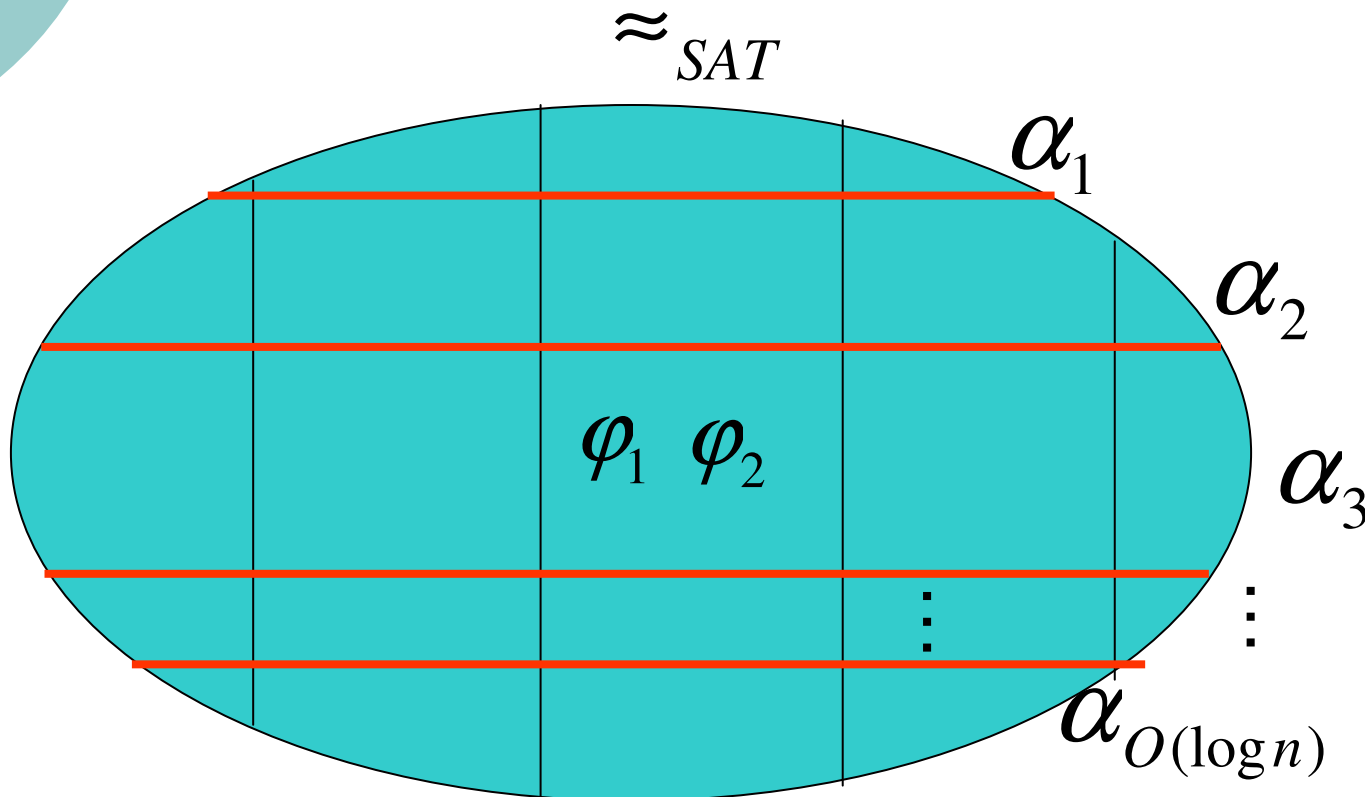
Can we use the ideas of [HKKN] for functions to get efficient almost private algorithms for search problems?

No.

[HKKN] use rounding of the result of a non-private approximation. Not clear how to generalize to search problems.

Almost Private Approximation for MAX-E3SAT

Every ϵ -approximator in the class \approx_{SAT} is divided into $O(\log n)$ sub-classes.





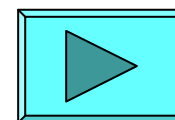
Lemma

There is a set of $O(\log n)$ assignments $\alpha_1, \dots, \alpha_{O(\log n)}$ such that **for every** 3SAT formula φ on n variables **there exists** an α_i that satisfies $7/8 - \varepsilon$ of the clauses in φ .

Proof:

Construct **almost 3-wise** independent variables x_1, \dots, x_n [NN, AGHP].

Number of assignments: $O(\frac{\log n}{\varepsilon})$.





Proof of Lemma 1 (cont.)

For every 3 random variables x_1, x_2, x_3
and every 3 Boolean values b_1, b_2, b_3 :
 $1/8 - \varepsilon < \Pr[x_1 = b_1 \wedge x_2 = b_2 \wedge x_3 = b_3] < 1/8 + \varepsilon$

Conclusion 1: For each clause C :

$\Pr[C \text{ is satisfied by } \alpha] > 7/8 - \varepsilon$
over the choice of α .

Conclusion 2: For every formula φ
there is an assignment that satisfies
 $7/8 - \varepsilon$ of its clauses.



Almost Private Approximation for MAX-3SAT

Thm 2: There exists a $(7/8 - \varepsilon)$ -approx algorithm for MAX-3SAT that leaks $O(\log \log n)$ bits.

Proof:

We use $\alpha_1, \dots, \alpha_{O(\log n)}$ from Lemma.

Given a formula φ return the first α_i that satisfies at least $(7/8 - \varepsilon)$ of the clauses in φ .



Solution-List Paradigm

- i A short list of solutions.
- i Every input has a good approximation in the list.
- i 2^k solutions \rightarrow algorithm leaks k bits



Further Results

- i Solution-list $n^{1-\varepsilon}$ -approximation algorithm for vertex cover that leaks $2n^\varepsilon$ bits.
- i **Impossibility result**
Any $n^{1-\varepsilon}$ -approximation algorithm for vertex cover must leak $\Omega(n^\varepsilon)$ bits.

Talk Overview



- i Background and Previous Work
- i Definitions for Search Problems
- i Impossibility Result for Vertex Cover
- i Algorithms that Leak (Little) Information
 - 1 Positive Result for MAX-3SAT
- i **Problems in P**
- i Conclusions and Open Problems





Problems in P – Private Computation

- Computation of a search problem in P might leak information.
- Many search problems in P have private algorithms (lex first):
 - perfect matching, shortest path, linear algebra, and more...
- Is there a private algorithm for every problem in P? **No!**



Problems in P - Private Computation

Let S be a search problem in P.

(Example: shortest-path)

Recall that $x \approx_S y$ if x and y have the same set of solutions.

For a private algorithm we require:

$$\mathbf{A}(x) \approx_c \mathbf{A}(y)$$

Is there a private algorithm for every problem in P? No!

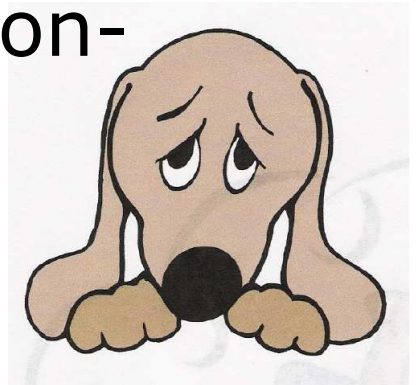
Impossibility result for a Problem in P

Input: $G = \langle V, E \rangle$, C , k

Output: If C is a clique of size k in G then output a clique of size k in G .

The problem is in P because C is a legal output.

A private algorithm implies a non-uniform algorithm for Clique.



Positive Results for Problems in P

Any problem S for which we can find:

- The lexicographically first solution
 - 1 $x \approx_S y$ implies x and y have the same lex first solution.
- A random solution
 - 1 $x \approx_S y$ implies that a random solution distributes identically for x and y .

Examples: perfect matching, shortest path, linear algebra, and more...





Discussion – Strength of Definition

We said the definition is minimal – good for impossibility results.

Is it **strong enough** for positive results?

Can returning the lex first solution be considered private?

What is the right **sufficient** definition? (work in progress...)

Talk Overview



- i Background and Previous Work
- i Definitions for Search Problems
- i Impossibility Result for Vertex Cover
- i Algorithms that Leak (Little) Information
 - 1 Positive Result for MAX-3SAT
- i Problems in P
- i **Conclusions and Open Problems**





Conclusions

- Defined private approximation of search problems
- **Impossibility result** for private approximation of vertex cover, max3SAT, and clustering problems
- Defined approximation algorithms for search problems with leakage
- **Positive result** for max3SAT
- Private computation of problems in P



Open Problems

- i More private approximation algorithms.
 - 1 Design algorithms that defeat solution list algorithms.
- i Private computation of problems in P .
 - 1 What is the right (sufficient) definition?
 - 1 What search problems admit efficient private computation?



köszönöm !תודה dĕkuji

mahalo 고맙습니다

thank you

merci 谢谢 *danke*

Eυχαριστώ شکرا

どうもありがとう *gracias*