Differential Privacy

M. Sc. Thesis

Lilla Tóthmérész

Advisor: Zoltán Király
Department of Computer Science

Eötvös Loránd University
Faculty of Natural Sciences
Budapest, 2013

Contents

Acknowledgement

I would like to thank my advisor Zoltán Király for the discussions we had, and for the many helpful ideas and suggestions he gave during my work.

1 Definitions

In recent years, technical development has enabled the collection of larger and larger databases of various kinds of data. There exist also many databases that contain sensitive (personal) data, for example hospital records, internet search information or the set of friends on different social sites. It is a common scenario that the release of a function/statistic on such data is socially beneficial. For example, the usage of hospital records greatly helps medical research. Nevertheless, if a database contains sensitive information, we would like to ensure, that the release of a function on that database does not leak too much information about the individuals. Differential privacy is a quite recent notion that tries to formalize this requirement.

How should privacy be defined? A natural definition would be to require that after learning the answer to a private query we have no extra knowledge about any individual in comparison with the earlier situation [3]. However, as it turns out, this is not achievable if we require the answer to have any nontrivial utility [3]. If we want the answer to have any utility, we must allow the leakage of some information. Therefore all we can hope for is to demand a bound on the extent of leakage. Differential privacy bounds the extent of knowledge we can learn about an individual if he joins the database. The philosophy is that instead of the real answer to a query, we output a random answer, such that by a small change in the database (someone joins or leaves), the distribution of the answer does not change much. Therefore, it is difficult to make the difference between the two input databases upon the knowledge of the answer using statistical methods. Intuitively, this means, that an individual does need to worry about joining the database. Naturally, we would like methods that add only small expected error, and therefore the output we get is still meaningful.

In the remaining of this thesis, we will use the following notations: We will denote the set of input databases by $\mathcal{D}$ and the output space of the query by R. We will generally denote the query function by F. Therefore $F : \mathcal{D} \to R$. We will denote the set of probability distributions over R by $\mu(R)$.

We will always have a distance function d on the set of databases. This distance measures what is an elementary change in a database that we would like to hide by the mechanism. This is most commonly the inclusion or the leaving out of one element, or the changing of one element.

Consider for example the function F, that on a database D outputs the number of records satisfying a given property P (we call these functions *counting queries* after [4]). For this problem, we take the following distance on databases: $d(D_1, D_2)$ is the minimal number of records one needs to delete from D_1 and need to add to it to get D_2. Specially $d(D_1, D_2) = 1$ if and only if D_1 contains one extra record in comparison with D_2 or vice versa. A differentially private mechanism with respect to this distance d will give statistically close answers for two databases that differ in one element.

As a second example, take the min cut problem. In the min cut problem, we have a graph with n vertices. The private input is a set of edges. We would like to get a cut of minimal size. Here, we say, that the distance of two graphs G_1 and G_2 is the cardinality of the symmetric difference of their edge sets. $d(G_1, G_2) = 1$ means that there is one extra edge in one graph compared to the other. If we use a differentially private mechanism to answer a min cut query, it will ensure that the answers are statistically close on two graphs that differ in one edge.

There is a third common choice for the distance function d: sometimes we measure the distance of databases in l_1 metric. Consider for example the case when we have $\mathcal{D} = \mathbb{R}^n$ (the databases contain n real numbers), and we want to compute the average. In this case, we say, that $d(D_1, D_2) = ||D_1 - D_2||_1$. The intuition behind using the l_1 metric is that we imagine, that our databases are the histograms of some data. A histogram is a mapping $H : \mathcal{D} \to \mathbb{N}^k$ where we have k disjoint properties, $P_1, \ldots P_k$, and $H(D)$ tells how many elements in the database have property $P_1, \ldots, P_k$ respectively. By disjoint, we mean that any element can satisfy at most one amongst properties $P_1, \ldots, P_k$. In this case, if we add a new record to the original database or remove one record, the histogram of the new database will have l_1-distance at most 1 from the histogram of the old database. If we apply a differentially private mechanism to compute the average, the outputs will be statistically close for two histograms with l_1 distance at most one. Therefore if the corresponding original databases differ in one element, the private averages of their histograms will be also statistically close to each other.

Let us now see the formal definition of differential privacy:

Definition 1.1 (Differential privacy [3]). *Let us be given a set $\mathcal{D}$ of databases with distance function d and an image set R. We call a randomized mechanism $M : \mathcal{D} \to \mu(R)$ ε- differentially private if for all $D_1, D_2 \in \mathcal{D}$ with $d(D_1, D_2) \leq 1$ and for all $S \subseteq R$ we have*

$$\mathbf{Pr}(M(D_1) \in S) \leq \exp(\varepsilon) \cdot \ \mathbf{Pr}(M(D_2) \in S).$$

Sometimes a weaker notion is used, where we let a δ additive change in the probabilities:

Definition 1.2 (Approximate differential privacy [5]). *Let us be given a set $\mathcal{D}$ of databases with distance function d and an image set R. We call a randomized mechanism $M : \mathcal{D} \to \mu(R)$ (ε, δ)- differentially private if for all $D_1, D_2 \in \mathcal{D}$ with $d(D_1, D_2) \leq 1$, and for all $S \subseteq R$ we have*

$$\mathbf{Pr}(M(D_1) \in S) \leq \exp(\varepsilon) \cdot \ \mathbf{Pr}(M(D_2) \in S) + \delta.$$

The amount of protection is determined by the parameter ε (and δ in the case of approximate privacy). The good choice of the value of ε is an important question for the applicability of the theory and also a question treated relatively rarely in the literature. We will not address this question in the thesis, either.

Consider the setting, where each element of a database is a private record of some individual. In this case, ε-differential privacy ensures, that even if the adversary knows each record in the database except for the record of a person x, he cannot learn much about the record of x. Whether he runs the algorithm with the correct record of x or with a fake record, he will get responses with statistically very close distributions. Therefore ε-differential privacy guarantees a strong protection against the enemy learning information based on others' data and the output.

However, differential privacy does not guarantee that the enemy cannot learn information about an individual based on the outcome of the mechanism, if he already had some auxiliary information about that individual. Also, we cannot hope for such a garantee. Consider for example the example of Terry Gross' height from [3]. Suppose that the height of a person is a private information. Our task is to release the average height of the population, and the enemy has the auxiliary information that Terry Gross is two centimeters shorter than the average height of the population. In this case, after learning the outcome of the query, the enemy is able to tell Terry Gross' height with the same amount of expected error as the mechanism gives. Therefore if we want to have a mechanism with small expected error, we cannot protect individuals against such threats.

There is another interesting interpretation of differential privacy, which is connected to mechanism design [10]. In mechanism design, an important notion is "truthfulness". A mechanism is truthful, if for each player, telling the truth is a dominant strategy. In other words, a player cannot achieve more income if he submits a fake input instead of the real one. Truthfulness is a desired property in the real life, and it is also a very appealing property in theory, because it simplifies the analysis of mechanisms. In the differential privacy framework, we can consider the elements of the databases as the inputs of the players. If an enemy wants to learn an element of the database and he knows all the other elements, he can run a query with some input on the unknown element. He can learn much information about that element, if the distribution of the output is very different in case of false inputs than on the real input. Differential privacy ensures, that the output distributions are close to each other whether the unknown input was given correctly or not. Translating this to the mechanism design framework, we get that a differentially private mechanism would give outputs where any event is at most $\exp(\varepsilon)$ times more likely if an individual lies than if he does not. Therefore we also have that the expected income of any individual is at most $\exp(\varepsilon)$ times more if he lies than if he does not lie. Also, the other players cannot loose much income because some other player lies. The expected income of a player when player x lies is at least $\exp(-\varepsilon)$ times the expected income when x does not lie. This property can be considered as an "approximate" truthfulness. An appealing property of this notion is that we automatically have a bound on the achievable extra income if a group of s players play together to maximize their income by falsifying their inputs. If a group S of $|S| = s$ players submit false input, each player achieves at most $\exp(s \cdot \varepsilon)$, and at least $\exp(-s \cdot \varepsilon)$ times the income he would get if the players in S would submit their real input. In contrast, real truthfulness can often only be proved if we do not allow collusions.

While we would like our mechanisms to be private, we would also like their expected error to be small. For being able to define the error of a mechanism, we need a score-function that tells us how good an output is for a given input.

Definition 1.3 (score-function)**.** *The score-function $q_F : \mathcal{D} \times R \to \mathbb{R}$ corresponding to function F determines how good a given output is for a given input. $q_F(D, r) \in \mathbb{R}$ means the value of output r on input D. (Intuitively it means how near is $F(D)$ to r.) Higher values mean better result. If the function F is clear from the context, we leave the subscript.*

$\mathrm{OPT}_F(D) := \max\{q_F(D, r) : r \in R\}$

For the applications, q can usually be defined naturally. Take for example the min cut problem. Here, we want to find the cut C that has minimum number of edges amongst the cuts of graph G. Let $q(G, C)$ is (-1) times the size of cut C in graph G. Therefore, the smallest a cut is, the better score it gets by the score function. The optimal cuts get the highest score, and we have $\mathrm{OPT}(G) = (-1)$ times the size of the min cut.

If a function takes its values from $\mathbb{R}^k$, then $q_F(D, r) = -||F(D) - r||$ is a natural value function where $||.||$ is a norm on $\mathbb{R}^k$. For $F =$ counting query, we can have $q(D, r) = -|F(D) - r|$. For $F =$ average, we can have $q(D, r) = -|(\sum_{x \in D} x)/|D| - r|$. In both cases, the nearest an output is to the real output of the function, the highest score they get.

Definition 1.4 (additive error-function)**.** *$h_F(x, r) = \mathrm{OPT}_F(x) - q_F(x, r)$.*

Definition 1.5 (multiplicative error-function)**.** *$h_{mult,F}(x, r) = \mathrm{OPT}_F(x)/q_F(x, r)$.*

Definition 1.6 (The additive error of a mechanism)**.** *For a function $F : \mathcal{D} \to R$ and an ε-differentially private mechanism $M : \mathcal{D} \to \mu(R)$,*

$$\mathrm{err}(M, F) = \max\{E_M(h_F(D, M(D))) : D \in \mathcal{D}\}$$

By E_M we mean the expectation over the randomness of M.

Definition 1.7 (The multiplicative error of a mechanism)**.** *We will only use this measure if the $q(D, r)$ are either all strictly positive or all strictly negative. For a function $F : \mathcal{D} \to R$ and an ε-differentially private mechanism $M : \mathcal{D} \to \mu(R)$,*

$$\mathrm{err}(M, F) = \max\{E_M(h_{mult,F}(D, M(D))) : D \in \mathcal{D}\}$$

The outline of this thesis is the following: In Section 2, we describe following the literature two generic methods for obtaining a differentially private mechanism: the exponential mechanism, and the K-norm mechanism. We also demonstrate some concrete examples for differentially private mechanisms. In this part, the described algorithms were previously known. An own result is the proof of Proposition 2.4, which is a statement about the accuracy of one of the algorithms from [7], that was not claimed in the original article. As another contribution, we point out, that in many of our example problems, K-norm mechanism actually coincides with the exponential mechanism.

In Section 3, we deal with cases where any differentially private algorithm needs to add unacceptably much noise to the data. In Subsection 3.1, we give a condition (Propositions 3.1 and 3.2) that guaranties that an ε-differentially private mechanism with constant expected error does not give a certain output with positive probability. Using this condition, we prove that for any ε, there exists no ε-differentially private mechanism computing the median problem, where the amount of expected error depends only on the cardinality of the input database. We also deduce lower bounds for the expected error needed to add to get a differentially private algorithm for the bounded median problem. These are own results inspired by some ideas in [7]. Later in this subsection, we shortly describe a framework called smoothed sensitivity (introduced in [11]), that tries to minimize the amount of expected error for the individual databases instead of the worst case. We point out, that the K-norm mechanism also works in a similar way, therefore it would worth to analyze also this method from this point of view.

In Subsection 3.2, we describe an interesting typical case where no non-trivial utility can be achieved with an ε-differentially private algorithm: the vertex cover problem. This example was pointed out by [7], and they also propose a modification of the problem, where differential privacy can be achieved with reasonable accuracy. We examine the relationship of their algorithm with the exponential mechanism, showing that they use a kind of "approximate" exponential mechanism as a subrutine. We also prove that this approximate exponential mechanism is ε-differentially private in itself in some cases (Theorem 3.1). We also point out a small error in the original proof of Proposition 3.7.

In Section 4, we demonstrate some results establishing lower bounds on the amount of error needed to add for certain problems to achieve ε-differential privacy. These are also previously known results. We try to emphasize the similarities of these proofs, and therefore we modified them as to all resemble the proof of Theorem 4.1 (which is from [8]).

2 Some methods for achieving privacy

2.1 The exponential mechanism

A mechanism achieves differential privacy by adding some random noise to the output. There are some general methods for doing this. Maybe the most common is the so called *exponential mechanism* [10].

Definition 2.1 (Exponential mechanism [10])**.** *The exponential mechanism $\mathcal{E}_q^\varepsilon$ belonging to a function F with score-function q gives an output r with the following density-function on an input D:*

$$f(r) = \frac{\exp(\varepsilon \cdot q(D,r))}{\int_R \exp(\varepsilon \cdot q(D,s))ds}$$

Suppose that we are given a function $F : \mathcal{D} \to R$ and a corresponding value function $q : \mathcal{D} \times R \to \mathbb{R}$. If $0 < \int_R \exp(\varepsilon \cdot q(D,s))ds < \infty$, then the above definition defines a randomized function from $\mathcal{D}$ to $\mu(R)$. (In typical cases, this integral is going to be finite.) We will show that this exponential mechanism provides differential privacy, where the privacy-parameter depends on the sensitivity of the score-function q.

Definition 2.2 (Sensitivity [10])**.**

$$\Delta = \Delta(q) = \max_{r \in R} \max_{d(D_1,D_2) \leq 1} |q(D_1,r) - q(D_2,r)|$$

Intuitively, the sensitivity means how large can be a change in the "goodness" of an output after an elementary change in the input database.

Theorem 2.1 ([10])**.** *The exponential mechanism corresponding to a function $F : \mathcal{D} \to R$, with score-function $q : \mathcal{D} \times R \to \mathbb{R}$ gives $2\varepsilon\Delta(q)$ - differential privacy.*

Proof. [10] The density function of $\mathcal{E}_q^\varepsilon(D)$ is

$$f_D(r) = \frac{\exp(\varepsilon q(D,r))}{\int_R \exp(\varepsilon q(D,s))ds}.$$

From this, we have

$$f_{D_1}(r) = \frac{\exp(\varepsilon q(D_1,r))}{\int_R \exp(\varepsilon q(D_1,s))ds} \leq \frac{\exp(\varepsilon(q(D_2,r)+\Delta))}{\int_R \exp(\varepsilon(q(D_2,s)-\Delta))ds} =$$

$$= \frac{\exp(\varepsilon\Delta) \cdot \exp(\varepsilon q(D_2,r))}{\exp(-\varepsilon\Delta) \cdot \int_R \exp(\varepsilon q(D_2,s))ds} \leq \exp(2\varepsilon\Delta) \cdot f_{D_2}(r).$$

Therefore for any $S \subseteq R$,

$$\mathbf{Pr}(\mathcal{E}_q^\varepsilon(D_1) \in S) = \int_S f_{D_1}(r)dr \leq$$

$$\leq \int_S \exp(2\varepsilon\Delta) \cdot f_{D_2}(r)dr = \exp(2\varepsilon\Delta)\ \mathbf{Pr}(\mathcal{E}_q^\varepsilon(D_2) \in S).$$

The other direction is implied by symmetry. □

If the output space is simple enough, sometimes we can efficiently sample from this distribution. If the size of R is polynomial in the size of the input, and $F(D)$ and $q(F(D),r)$ is computable in polynomial time, then the exponential mechanism gives a differentially private algorithm with polynomial running time. If the size of R is exponential, which is a common case for combinatorial problems, we need various tricks to achieve a polynomial time algorithm (we will see examples in Section 2.3.2).

2.2 The accuracy of the exponential mechanism

Gupta et al. [7] gives a theorem about the accuracy of the exponential mechanism for finite R stating that the probability of obtaining a highly suboptimal output is exponentially small.

Theorem 2.2 ([7]). *Let R be finite, and denote by $R_{OPT}(D)$ the set of optimal outputs for input D: $R_{OPT}(D) = \{r \in R : \ q(D,r) = \mathrm{OPT}(D)\}$. Then*

$$\mathbf{Pr}\Big(q(D, \mathcal{E}_q^\varepsilon(D)) < OPT(D) - \frac{ln(|R|/|R_{OPT}(D)|)}{\varepsilon} - \frac{t}{\varepsilon}\Big) \leq \exp(-t).$$

Proof. [7] If for an $r \in R$, $\quad q(D,r) < OPT(D) - \frac{ln(|R|/|R_{OPT}(D)|)}{\varepsilon} - \frac{t}{\varepsilon}$, then

$$\mathbf{Pr}(\mathcal{E}_q^\varepsilon(D) = r) = \frac{\exp(\varepsilon q(D,r))}{\sum_{s\in R}\exp(\varepsilon q(D,s))} \leq$$

$$\leq \frac{\exp(\varepsilon \cdot OPT(D) - ln(|R|/|R_{OPT}(D)|) - t)}{\sum_{s\in R}\exp(\varepsilon q(D,s))} =$$

$$= \frac{\exp(\varepsilon \cdot OPT(D) - t) \cdot \frac{|R_{OPT}(D)|}{|R|}}{\sum_{s\in R}\exp(\varepsilon q(D,s))} \leq$$

$$\leq \frac{\exp(\varepsilon \cdot OPT(D) - t) \cdot \frac{|R_{OPT}(D)|}{|R|}}{|R_{OPT}(D)| \cdot \exp(\varepsilon \cdot OPT(D))} = \frac{\exp(-t)}{|R|}$$

There are at most $|R|$ such r, therefore the probability that we get an r with $q(D,r) < OPT(D) - \frac{ln(|R|/|R_{OPT}(D)|)}{\varepsilon} - \frac{t}{\varepsilon}$ is at most $\exp(-t)$. □

2.3 Examples for the usage of the exponential mechanism

2.3.1 A simple special case: $R = \mathbb{R}^d$: The Laplacian mechanism

A simple case is, if the function F takes its values from $\mathbb{R}^d$, i.e. $R = \mathbb{R}^d$. A natural score-function in this case is $q(D,r) = -||F(D) - r||$ for some norm on $\mathbb{R}^d$. We have $\Delta(q) = \max_{r\in\mathbb{R}^n}\max_{d(D_1,D_2)\leq 1}\Big(\Big| - ||F(D_1) - r|| + ||F(D_2) - r||\Big|\Big) \leq \max_{d(D_1,D_2)\leq 1}\{||F(D_1) - F(D_2)||\} =: \Delta(F)$ from the triangle-inequality.

If we would like an ε-differentially private mechanism, we can apply the exponential mechanism with $\varepsilon' = \varepsilon/2\Delta(F)$. But note, that in this special case, $\varepsilon' = \varepsilon/\Delta(F)$ also gives an ε-differentially private algorithm, because we know, that the denominator $\int_R \exp(\varepsilon q(D,s))ds = \int_{\mathbb{R}^d}\exp(-\varepsilon||F(D) - s||)ds$ which is the same value for each $D \in \mathcal{D}$.

The density function on input D is going to be

$$f_D(r) = \frac{\exp(-\varepsilon'||F(D) - r||)}{\int_R \exp(-\varepsilon'||F(D) - s||)ds}.$$

In one dimension, with $||.|| = |.|$, this means, that we add error to $F(D)$ with Laplace (symmetric exponential) distribution Laplace(0, $\Delta(f)/\varepsilon$). This special case is the Laplacian mechanism, introduced by [6].

The expected (additive) error of the mechanism is $E(|\text{Laplace}(0, \Delta(f)/\varepsilon)|) = \Delta(f)/\varepsilon$.

In $\mathbb{R}^d$, if we choose $||.||$ to be $||.||_1$ then we can add error with Laplace distribution such that we add error independently to each coordinate with distribution Laplace(0, $\Delta(f)/\varepsilon$).

The expected error is then the sum of the expected error of the coordinates, so $\frac{d \cdot \Delta(f)}{\varepsilon}$.

As we can sample efficiently from Laplace distribution, in these cases, if we have an efficient algorithm for computing F, we also have an efficient algorithm for computing the differentially private version. Let us see some examples.

Counting queries In counting queries,

$$F(D) = \text{ the number of elements in the database that have a given property } P.$$

By adding or deleting one element of the database, F can change by at most 1, therefore $\Delta(F) = 1$. The score-function is $q(x, r) = -|F(x) - r|$. If we allow the mechanism to output any real number (not only integers), than we can apply the Laplacian mechanism [4].

The expected (additive) error is $E(|\text{Laplace}(0, 1/\varepsilon)|) = 1/\varepsilon$.

Histogram queries Histogram queries are similar to counting queries. Here, we have d disjoint properties, $P_1, \ldots P_d$. By disjoint, we mean, that any element in the universe can satisfy at most one amongst properties $P_1, \ldots P_d$. $F(D)$ is a d-dimensional vector that tells how many elements in the database have property $P_1, \ldots, P_d$ respectively. The score-function is $q(D, r) = -||F(D) - r||_1$. Here, once more, we have $\Delta(F) = 1$, because the addition or the removal of one element from the database can only affect the number of elements for one property, because of the disjointness of the properties. (This is the point where histogram queries differ from the union of d counting queries.) We can apply the Laplacian mechanism [4], and get expected error $\frac{d}{\varepsilon}$.

Linear mappings Let F be a linear mapping from $\mathbb{R}^n$to $\mathbb{R}^d$. For this case, we measure the distance between databases in the l_1 metric (or in other words, we suppose that the input databases are histograms of some data). We have $\Delta(F) = \max\{||F(x) - F(y)|| : \; ||x - y||_1 \leq 1\} = \max\{||F(x - y)|| : \; ||x - y||_1 \leq 1\} = ||F||$, therefore if we measure the error in l_1 norm, the expected (additive) error is $\frac{d||F||}{\varepsilon}$.

2.3.2 Combinatorial problems

In combinatorial problems, our task is much more difficult, because the output space typically has a more difficult structure and so we cannot sample from the distribution given by the exponential mechanism in such a simple way as for the Laplacian case. Very often we look for some optimal subset within the input. Therefore, typically the number of possible outputs is exponential in the size of the input. In these cases, the direct application of the exponential mechanism would yield an algorithm with exponential running time. However, there are some tricks we can use. If we can generate (in a private manner) a set of possible outputs with polynomial size such that the exponential mechanism would sample amongst them with probability $1-\delta$ (this means for example generating all the near-optimal inputs), then by sampling amongst only them, we get an (ε, δ)-differentially private mechanism. Another possibility is to find an algorithm that selects from a polynomial size set in each step, and convert these steps to be private (for example with the exponential mechanism). Let us see an example for both solutions from the paper of Gupta et al. [7].

Min Cut Gupta et al. [7] proposes an algorithm for the private min cut problem, that applies the first trick. They give an $(\varepsilon, O(1/n^2))$-private algorithm for min cut that gives a cut with expected size at most $\mathrm{OPT} + O(\frac{\ln n}{\varepsilon})$.

In the min cut problem, we are given an undirected graph G, and we look for a cut of minimal size. We suppose that the number of nodes in the graph is a fixed number n (or in another way of seeing it, it is a public parameter). The private input is the edge set. The mechanism will output a set $C \subseteq V$ that defines the cut $(C, V \setminus C)$.

Notation. *Let us denote the size of a cut $(C, V \setminus C)$ in a graph $G = (V, E)$ by $Cost(G, C)$.*

Let us denote the size of the min cut in G by $c(G)$.

In this problem, we have $q(G, C) = -Cost(G, C)$.

This q has sensitivity 1, because by changing one edge in the graph, the size of any cut changes by at most one. So the direct application of the exponential mechanism for sampling amongst all the cuts in the graph would be a 2ε-differentially private algorithm. However, there are exponentially many cuts in a graph, so this would not be an efficient algorithm.

Algorithm 1 Private min cut, exponential mechanism (inefficient) [7]

```
**Input:** $G = (V, E), \varepsilon$
choose any cut $C_i$ with probability proportional to $\exp(-\varepsilon \cdot Cost(G, C_i))$
**Output:** the cut $C_i$
```

The algorithm of Gupta et al. [7] employs Karger's algorithm [9] and some of his results concerning min cuts. For this reason, let us repeat these results. Suppose that the size of the minimum cut in G is c. Karger's algorithm is a randomized algorithm, that on input $G = (V, E)$, outputs a cut $(C, V \setminus C)$ such that for each cut $(C, V \setminus C)$ of minimal size,

$$\mathbf{Pr}(\text{the output is } C) \geq \frac{1}{n^2},$$

and also for each cut $(C, V \setminus C)$ of size at most kc

$$\mathbf{Pr}(\text{the output is } C) \geq \frac{1}{n^{2k}}.$$

This algorithm has two important consequences:

Corollary 2.1. *[9] The number of cuts of size at most kc is at most n^{2k}.*

Corollary 2.2. *[9] If we employ Karger's algorithm independently n^{2k+1} times, we get each cut of size at most kc, except with exponentially small probability.*

Proof. [9] Fix a cut $(C, V \setminus C)$ of size at most kc. In each run, the probability that we do not get C is at most $(1 - \frac{1}{n^{2k}})$. The probability that we do not see it in either of the runs is therefore at most

$$(1 - \frac{1}{n^{2k}})^{n^{2k+1}} \sim \frac{1}{e^n}.$$

□

Before describing the algorithm of Gupta et al., let us point out an interesting phenomenon in connection with Algorithm 1 (also from [7]). If we would like to approximate the expected error of the above inefficient algorithm, we would need to bound the number of cuts with size $c(G) + k$ for a given k. From Corollary 2.1, we have a bound on the number of cuts of size at most $k \cdot c(G)$, however, if $c(G)$ is small, this gives a very weak bound on the number of cuts with size $c(G) + k$. This suggests, that we can prove a better bound on the accuracy of the inefficient algorithm if before running the exponential mechanism, we raise the size of the minimal cut to a sufficiently large size compared to n (in a differentially private manner). This is indeed what [7] does: they raise the size of the minimal cut to approximately $\frac{4 \ln n}{\varepsilon}$. Then they can prove that the (still inefficient) algorithm gives a cut with expected size $c(G) + O(\frac{\ln n}{\varepsilon})$.

Algorithm 2 Private min cut, improved exponential mechanism (inefficient) [7]

```
Input: $G = (V, E), \varepsilon$
fix: $\emptyset = H_0 \subset H_1 \subset \ldots \subset H_{\binom{n}{2}}$
choose $H_j$ with probability proportional to $\exp(-\varepsilon|c(G \cup H_j) - \frac{8 \ln n}{\varepsilon}|)$
choose any cut $C_i$ with probability proportional to $\exp(-\varepsilon \cdot Cost(G \cup H_j, C_i))$
Output: the cut $C_i$
```

To raise the size of the min cut to approximately $\frac{4 \ln n}{\varepsilon}$ in a differentially private manner, they fix an order of strictly increasing sets of edges: $\emptyset = H_0 \subset H_1 \subset \ldots \subset H_{\binom{n}{2}}$ (by edges, here we mean the edges of the complete graph, not necessarily edges within E). Then they choose a set H_i amongst these sets with probability proportional to $\exp(-\varepsilon|c(G \cup H_i) - \frac{8\ln n}{\varepsilon}|)$. Thus the new graph with raised min cut size is going to be $G' = (V, E \cup H_i)$. This is also the application of the exponential algorithm, where we would like to minimize the distance of the size of the min cut from $\frac{8\ln n}{\varepsilon}$.

Proposition 2.1. *Algorithm 2 preserves* (4ε)*-differential privacy.*

We note that Gupta et al. claims this without proof with 2ε-differential privacy.

Proof. Let us denote the mechanism given by Algorithm 2 by M.

$\Delta(|c(G \cup H_i) - \frac{8\ln n}{\varepsilon}|) = 1$, therefore the exponential mechanism of the first phase is 2ε-differentially private. Let us denote the mechanism of the first phase by $PH1$. We have

$$\mathbf{Pr}(PH1(G) = i) \leq \exp(2\varepsilon)\ \mathbf{Pr}(PH1(G') = i),$$

if G and G' differ in one edge.

For a fixed i, by changing one edge in the graph G, $Cost(G \cup H_j, C_i)$ changes by at most 1 for each cut C_i. So for each $S \subsetneqq 2^V$,

$$\mathbf{Pr}(M(G) \in S \mid PH1(G) = i) \leq \exp(2\varepsilon)\ \mathbf{Pr}(M(G') \in S \mid PH1(G') = i)$$

where G and G' differ in one edge.

Therefore

$$\mathbf{Pr}(M(G) \in S) = \sum_{i=1}^{\binom{n}{2}} \mathbf{Pr}(M(G) \in S \mid PH1(G) = i)\ \mathbf{Pr}(PH1(G) = i) \leq$$

$$\leq \sum_{i=1}^{\binom{n}{2}} \exp(2\varepsilon)\ \mathbf{Pr}(M(G') \in S \mid PH1(G') = i) \exp(2\varepsilon)\ \mathbf{Pr}(PH1(G') = i) =$$

$$= \exp(4\varepsilon)\ \mathbf{Pr}(M(G') \in S)$$

□

We include here the statement about the accuracy of this mechanism.

Proposition 2.2. *[7] For any graph* G *and* $\varepsilon < 1$*, the expected size of the cut given by Algorithm 2 is at most* $c(G) + O(\frac{\ln n}{\varepsilon})$.

Proof. In the proof, we follow [7]. First we claim, that for $j = PH1(G)$,

$$\mathbf{Pr}\Big[\frac{4\ln n}{\varepsilon} < c(G \cup H_j) < c(G) + \frac{12\ln n}{\varepsilon}\Big] \geq 1 - \frac{1}{n^2}. \tag{1}$$

If $c(G) \geq \frac{8\ln n}{\varepsilon}$, then the optimal choice in phase 1 is H_0 (and this is the only optimal choice). From Theorem 2.2,

$$\mathbf{Pr}\Big[\Big|c(G \cup H_{PH1(G)}) - \frac{8\ln n}{\varepsilon}\Big| > \Big|c(G) - \frac{8\ln n}{\varepsilon}\Big| + \ln\binom{n}{2} + \frac{2\ln n}{\varepsilon}\Big] \leq \exp(-\frac{2\ln n}{\varepsilon}).$$

Since $c(G\cup H_{PH1(G)})-\frac{8\ln n}{\varepsilon}$ and $c(G)-\frac{8\ln n}{\varepsilon}$ are both positive, we have:

$$\mathbf{Pr}(c(G\cup H_{PH1(G)})>c(G)+\frac{4\ln n}{\varepsilon})\le\frac{1}{n^2}.$$

This implies (1) in this case (since we always have $c(G\cup H_{PH1(G)})\ge c(G))>\frac{4\ln n}{\varepsilon})$.

If $c(G)<\frac{8\ln n}{\varepsilon}$, then there is an index i such that $c(G\cup H_i)=\frac{8\ln n}{\varepsilon}$. This is true because for the consecutive i, the value of $c(G\cup H_i)$ grows by at most one, and since $G\cup H_{\binom{n}{2}}$ is a complete graph, $c(G\cup H_{\binom{n}{2}})=n-1>\frac{8\ln n}{\varepsilon}$ (for large n; since ε is constant). In this case we have:

$$\mathbf{Pr}\Big[|c(G\cup H_{PH1(G)})-\frac{8\ln n}{\varepsilon}|>\frac{4\ln n}{\varepsilon}\Big]\le\frac{1}{n^2}$$

that again implies (1) since $c(G)\ge 0$.

Now fix a j such that $c(G\cup H_j)>\frac{4\ln n}{\varepsilon}$. Let us denote the number of cuts C with $Cost(G\cup H_j,C)\le c(G\cup H_j)+t$ by c_t.

$$c(G\cup H_j)+t=(1+t/c(G\cup H_j))c(G\cup H_j)$$

$\Rightarrow$ by Corollary 2.1, $c_t\le n^{2(1+t/OPT(G\cup H_j))}=n^2n^{\frac{2\varepsilon t}{4\ln n}}=n^2\exp(\frac{\varepsilon t\ln n}{2\ln n})=n^2\exp(\frac{\varepsilon t}{2})$.

There exists a cut of size $c(G\cup H_j)$, therefore the normalization sum in the exponential mechanism is at least $\exp(-\varepsilon\cdot c(G\cup H_j))$. So for a cut C_i, with $Cost(G\cup H_j,C_i)\ge c(G\cup H_j)+t$,

$$\mathbf{Pr}(M(G)=C_i|PH1(G)=j)\le\frac{\exp(-\varepsilon(c(G\cup H_j)+t))}{\exp(-\varepsilon c(G\cup H_j))}=\exp(-\varepsilon t).$$

For the expected error, we have: $E(c(G\cup H_{PH1(G)}))\le(1-\frac{1}{n^2})\frac{12\ln n}{\varepsilon}+\frac{1}{n^2}n^2=O(\frac{\ln n}{\varepsilon})$, so in the first phase, we add $O(\frac{\ln n}{\varepsilon})$ expected error.

In the second phase: For a j such that $\frac{4\ln n}{\varepsilon}<c(G\cup H_j)<c(G)+\frac{12\ln n}{\varepsilon}$, we have

$$P(Cost(G\cup H_j,M(G))>c(G\cup H_j)+b\mid PH1(G)=j)\le$$

$$\le\sum_{t>b}(\exp(-\varepsilon t)(c_t-c_{t-1}))\le-\exp(-\varepsilon(b+1))c_b+(1-\exp(-\varepsilon))\sum_{t>b}\exp(-\varepsilon t)c_t\le$$

$$\le(1-\exp(-\varepsilon))\sum_{t>b}\exp\Big(-\frac{\varepsilon t}{2}\Big)n^2=\sum_{t>b}\Big[\exp\Big(-\frac{\varepsilon t}{2}\Big)-\exp\Big(-\frac{\varepsilon(t+2)}{2}\Big)\Big]n^2=$$

$$=\Big[\exp\Big(-\frac{\varepsilon(b+1)}{2}\Big)+\exp\Big(-\frac{\varepsilon(b+3)}{2}\Big)\Big]n^2=\Big[\exp\Big(-\frac{\varepsilon}{2}\Big)+\exp\Big(-\frac{3\varepsilon}{2}\Big)\Big]\exp\Big(-\frac{\varepsilon b}{2}\Big)n^2.$$

For $b=\frac{8\ln n}{\varepsilon}$, this gives

$$P(Cost(G\cup H_j,M(G))>c(G\cup H_j)+b\mid PH1(G)=j)\le$$

$$\leq \frac{[\exp(-\frac{\varepsilon}{2}) + \exp(-\frac{3\varepsilon}{2})]}{n^2} \leq \frac{2}{n^2}.$$

Summing up the results: With probability at least $1-\frac{1}{n^2}$, in the first phase, we get a j such that $\frac{4\ln n}{\varepsilon} < c(G \cup H_j) < c(G) + \frac{12\ln n}{\varepsilon}$. In this case, we get a cut with expected additive error $O(\frac{\ln n}{\varepsilon})$, because with probability at least $1 - \frac{2}{n^2}$, we get at most $\frac{8\ln n}{\varepsilon}$ error in addition to $c(G \cup H_j)$ which is at most $c(G) + \frac{12\ln n}{\varepsilon}$. With the remaining $\frac{2}{n^2}$ probability, we get a cut with error at most $n^2/2$, so this adds at most 1 to the expected error.

With probability at most $\frac{1}{n^2}$, the first phase does not give us a nice H_j, but also in this case, the error is at most n^2, so this case adds at most 1 to the expected additive error.

□

Now, we can describe the efficient algorithm. Interestingly, the raising of the size of the min cut will be crucial in two more points in the efficient algorithm.

The idea of the efficient algorithm is that the reason why the exponential mechanism was inefficient for min cut is that there are exponentially many cuts in a graph. But since there are only polynomially many min cuts, we can try to generate a polynomial number of "small cuts", and sample amongst them using the exponential mechanism. Gupta et al. employ Karger's algorithm for generating the cuts of size at most $3c(G)$. For the exponential mechanism, we can more easily handle the probabilities of choosing items with a certain additive error in comparison with the optimum. In Karger's algorithm, we can deal with the probabilities of outputting cuts with a certain multiplicative error. Therefore we will again need the trick of raising the value of the min cut for being able to proove the $(\varepsilon, \frac{1}{n^2})$-privacy of the algorithm.

Algorithm 3 Private min cut, efficient algorithm [7]

```
  **Input:** $G = (V, E), \varepsilon$
  fix: $\emptyset = H_0 \subset H_1 \subset \ldots \subset H_{\binom{n}{2}}$ strictly increasing sets of edges
  choose $H_j$ with probability proportional to $\exp(-\varepsilon|c(G \cup H_j) - \frac{8\ln n}{\varepsilon}|)$
  run Karger's algorithm $n^7$ times on $G \cup H_j \rightarrow$ cuts $\mathcal{C} = \{C_1, \ldots, C_{n^7}\}$
  choose a $C_i \in \mathcal{C}$ with probability proportional to $\exp(-\varepsilon \cdot Cost(G \cup H_j, C_i))$
  **Output:** the cut $C_i$
```

The algorithm will be the following (Algorithm 3): We raise the value of the min cut like in Algorithm 2. We run Karger's algorithm n^7 times. In each run i, we get a cut C_i. After this, we employ the exponential mechanism for choosing amongst these cuts. This indeed gives a polynomial algorithm.

Proposition 2.3. *[7] Algorithm 3 preserves $(4\varepsilon, O(\frac{1}{n^2}))$-differential privacy.*

Proof. We again follow [7]. We have already proved, that if we sampled amongst all the cuts and not only amongst the cuts generated by Karger's algorithm, then we would get a 4ε-differentially private algorithm (Algorithm 2). Imagine that we run Karger's algorithm n^7 times, but we sample amongst all the cuts. Then the probability that we get a cut that was not generated by the runs of Karger's algorithm is $O(\frac{1}{n^2})$:

For a cut C with $Cost(G \cup H_j, C) < 3c(G \cup H_j)$, the probability that we did not get it by n^7 runs of Karger's algorithm is is exponentially small.

For a cut C with $Cost(G \cup H_j, C) > 3c(G \cup H_j)$: The probability that $c(G \cup H_j) \leq \frac{4 \ln n}{\varepsilon}$ is at most $\frac{1}{n^2}$. If $c(G \cup H_j) > \frac{4 \ln n}{\varepsilon}$, then for the cuts with $Cost(G \cup H_j, C) > 3c(G \cup H_j)$, $Cost(G \cup H_j, C) > c(G \cup H_j) + \frac{8 \ln n}{\varepsilon}$. As we have shown in the proof of Proposition 2.2, the probability of getting such a cut (conditioned on having a j with $c(G \cup H_j) > \frac{4 \ln n}{\varepsilon}$) is also $O(\frac{1}{n^2})$.

Sampling amongst the cuts generated by the runs of Karger's algorithm from the distribution given by the exponential mechanism on that restricted set is equivalent to the following process: We sample amongst all the cuts using the exponential mechanism on the whole set, but if we get a cut that was not generated by the runs of Karger's algorithm, then we start the process once more. This can be seen from the following train of thought: Let us denote the weights of the cuts generated by the runs of Karger's algorithm by $a_1, \ldots, a_k$ respectively, and the the weights of the other cuts by $b_1, \ldots, b_l$ respectively. $\sum_j a_j =: A$, $\sum_j b_j =: B$. Then for each a_i, the probability of sampling a_i amongst $\{a_1, \ldots, a_k\}$ is $\frac{a_i}{A}$ and the probability of sampling it from the second process is

$$\frac{a_i}{A+B} + \frac{a_i}{A+B} \cdot \frac{B}{A+B} + \ldots = \frac{a_i}{A+B} \Big(\sum_{s=0}^{\infty} \Big(\frac{B}{A+B} \Big)^s \Big) = \frac{a_i}{A+B} \cdot \frac{A+B}{A} = \frac{a_i}{A}.$$

Since in the second process, the probability that we do not get a cut from the generated cuts in the first sampling is at most $O(\frac{1}{n^2})$, the probability of any event can change at most by $O(\frac{1}{n^2})$ compared to the case of sampling amongst all the cuts. Therefore Algorithm 3 has $(4\varepsilon, O(\frac{1}{n^2}))$-differential privacy. □

Proposition 2.4. *The expected cost of Algorithm 3 is at most $c(G) + O(\frac{\ln n}{\varepsilon})$.*

Proof. Once more we can think of Algorithm 3 as sampling amongst all the cuts but restarting when it gives a cut not generated by the runs of Karger'a algorithm. The probability that the algorithm needs a second sampling is at most $O(\frac{1}{n^2})$. In that case, it gives a cut of size at most n^2. Otherwise it gives the same cut as Algorithm 2 would give. Therefore the two algorithms give different cuts on an event of measure at most $O(\frac{1}{n^2})$, and in these cases the difference in the size of the output cut is at most n^2. Therefore the expected size of the output differs by $O(1)$. As Algorithm 2 has expected cost $c(G) + O(\frac{\ln n}{\varepsilon})$, this is also true for Algorithm 3. □

k-Median For the k-median problem, Gupta et al. [7] gives a polynomial algorithm using the second trick: They start from an arbitrary set with k elements and privately improve it. In these improving steps, they only need to select among a polynomial number of objects.

The setting in the k-median problem is the following: We have a set of points V with a metric $d : V \times V \to \mathbb{R}$. The private input is a set of demand points $D \subseteq V$. We need to select a set $F \subseteq V$, $|F| = k$ of facilities, such that $cost(F) := \sum_{v \in D} d(v, F)$ is minimal.

For an input D, let us denote $\min\{cost(F) : |F| = k\}$ by $c(D)$.

The solution of [7] uses the theorem of Arya et al. [1] :

Theorem 2.3. *[1] For any set $F \subseteq V$, $|F| = k$, there exists a set of k swaps $(x_1, y_1), \ldots, (x_k, y_k)$ such that $\sum_{i=1}^{k} (cost(F) - cost(F - x_i + y_i)) \geq cost(F) - 5c(D)$.*

Corollary 2.3. *[7] For any set $F \subseteq V$, $|F| = k$, there exists a swap (x, y) such that $cost(F) - cost(F - x + y) \geq \frac{1}{k}(cost(F) - 5c(D))$.*

The algorithm will start with an arbitrary F and in T steps, it selects an improving swap using the exponential mechanism. Then, at the end, it selects the approximately best k-median seen during the T steps, again using the exponential mechanism.

Let $\Delta = diam(V)$. Note that by changing an element in D, the cost of F can change by at most Δ thus we have $\Delta(q) \leq \Delta$.

Algorithm 4 k-median algorithm [7]

```
**Non-private input:** $V$, $d$, $k$, $\varepsilon$. **Private input:** $D \subseteq V$.
$T := 6k \ln n$, $\varepsilon' := \varepsilon/(2\Delta(T+1))$.
Let $F_1 \subseteq V$ be arbitrary such that $|F_1| = k$.
**for** $i = 1, \ldots, T$ **do**
    Select $(x, y) \in F_i \times (V \setminus F_i)$ with probability proportional to $\exp(-\varepsilon' \cdot cost(F_i - x + y))$.
    $F_{i+1} := F_i - x + y$
**end for**
Select $j$ from $\{1, \ldots T\}$ with probability proportional to $\exp(-\varepsilon' \cdot cost(F_j))$.
**Output:** $F_j$
```

Proposition 2.5. *[7] Algorithm 4 provides ε-differential privacy.*

Proof. We follow [7]. Since q has sensitivity Δ, the first swapping step has $2\varepsilon'\Delta = \varepsilon/(T+1)$-differential privacy. In the second step, for each fixed F, the probability that we choose a given swap changes by at most a factor of $\exp(2\varepsilon'\Delta)$. Also, the probability that we get a given $F_1 = F$ changes by at most a factor of $\exp(2\varepsilon'\Delta)$ from the $2\varepsilon'\Delta$-differential privacy of the first step. Therefore the second step is $4\varepsilon'\Delta = 2\varepsilon/(T+1)$-private. Similarly, the third step is $3\varepsilon/(T+1)$-private and so on. The T^{th} step is $\varepsilon T/(T+1)$-differentially private. When finally we select amongst the F_i, the cost of any F_i changes by at most Δ upon changing an element in D. So for any given set $\{F_1, \ldots, F_T\}$, the probability that we select a given set changes by at most a factor of $\exp(2\varepsilon'\Delta)$. We have

$$\mathbf{Pr}[M(D) = F] = \sum_{i=1}^{T} \mathbf{Pr}[M(D) = F \mid F = F_i] \, \mathbf{Pr}_D[F = F_i] \leq$$

$$\leq \sum_{i=1}^{T} \exp(2\varepsilon'\Delta) \, \mathbf{Pr}[M(D') = F \mid F = F_i] \exp(2i\varepsilon'\Delta) \, \mathbf{Pr}_{D'}[F = F_i] \leq$$

$$\leq \exp(\varepsilon) \sum_{i=1}^{T} \mathbf{Pr}[M(D') = F \mid F = F_i] \, \mathbf{Pr}_{D'}[F = F_i] = \exp(\varepsilon) \, \mathbf{Pr}[M(D') = F]$$

□

Proposition 2.6. *[7] Except with probability $O(1/poly(n))$, Algorithm 4 outputs a solution of cost at most $6c(D) + O(\frac{\Delta k^2 \ln^2 n}{\varepsilon})$.*

Proof. [7] We use Corollary 2.3 and Theorem 2.2. If for an i we have $cost(F_i) \geq 6c(D) + \frac{24k \ln n}{\varepsilon'}$, then there is a swap, that reduces the cost by

$$\frac{cost(F_i) - 5c(D)}{k} = \frac{cost(F_i) - 5c(D) - \frac{20k \ln n}{\varepsilon'} + \frac{20k \ln n}{\varepsilon'}}{k} \geq \frac{cost(F_i)}{6k} + \frac{20 \ln n}{\varepsilon'}.$$

Since there are at most n^2 possible swaps, from Theorem 2.2, with probability at least $1 - \frac{1}{n^2}$, we select a swap with distance at most $\frac{4 \ln n}{\varepsilon'}$ from the optimal. Therefore with probability at least $1 - \frac{1}{n^2}$, we select a swap such that $cost(F_{i+1}) \leq (1 - \frac{1}{6k}) cost(F_i)$.

$cost(F_0) \leq n\Delta$. If we had $cost(F_i) \geq 6c(D) + \frac{24k \ln n}{\varepsilon'}$ in each phases, with probability at least $1 - \frac{T}{n^2}$, the cost would decrease by a factor of $1 - \frac{1}{6k}$ in each step. $n\Delta(1 - \frac{1}{6k})^T \leq \Delta \leq \frac{24k \ln n}{\varepsilon'} = \frac{48k\Delta \ln n}{\varepsilon}$ therefore with probability at least $1 - \frac{T}{n^2}$, we have an i such that $cost(F_i) < 6c(D) + \frac{24k \ln n}{\varepsilon'}$.

In the final step, we choose an F_j within error of $\frac{4 \ln n}{\varepsilon'}$ of $\min_i\{cost(F_i)\}$ with probability at least $1 - \frac{1}{n^2}$. So with probability $1 - \frac{1}{n^2}$, we get an output F with $cost(F) < 6c(D) + \frac{28k \ln n}{\varepsilon'}$. Substituting $\varepsilon' = \varepsilon/2\Delta(T+1) = \varepsilon/2\Delta(6k \ln n + 1)$, this means $cost(F) < 6c(D) + O(\frac{\Delta k^2 \ln^2 n}{\varepsilon'})$ with probability $1 - \frac{1}{n^2}$. For any F, $cost(F) \leq n\Delta$. Therefore the remaining cases add at most $\frac{\Delta}{n}$ expected error. □

2.4 The K-norm mechanism

The K-norm mechanism [8] is a version of the exponential mechanism, but where we do not use the function q (which is to be maximized) to determine the probabilities, but another function q'. Hardt and Talwar introduced it in [8] for the case of $\mathbb{R}^n \to \mathbb{R}^d$ linear mappings, and showed, that in that case it can give better expected error then the exponential mechanism. We first repeat their definition and results, then write about the relationship between the exponential and the K-norm mechanisms.

Definition 2.3 (Minkowski-norm)**.** *For a $K \subseteq \mathbb{R}^d$ centrally symmetric convex set, the Minkowski-norm defined by K is the following:*

$$||x||_K = \inf\{r : x \in rK\}.$$

Note that the Minkowski-norm is indeed a norm.

Let $F : \mathbb{R}^n \to \mathbb{R}^d$ be a linear mapping, and let $K = FB_1^n$, where B_1^n is the l_1-unit ball of $\mathbb{R}^n$. From the linearity of F, K is a centrally symmetric convex set. Then the K-norm mechanism for F is the following:

Definition 2.4 (K-norm mechanism for $\mathbb{R}^n \to \mathbb{R}^d$ linear mappings [8])**.** *For a given $\varepsilon > 0$, the K-norm mechanism for F is the mechanism, that on input $x \in \mathbb{R}^n$ returns a random output $a \in \mathbb{R}^d$ with probability density function*

$$f_x(a) = Z^{-1} \exp(-\varepsilon||Fx - a||_k),$$

where $Z = \int_{\mathbb{R}^d} \exp(-\varepsilon||Fx - a||_K)da = \Gamma(d+1)\mathrm{Vol}(\varepsilon^{-1}K)$.

Proposition 2.7. *[8] The K-norm mechanism for $\mathbb{R}^n \to \mathbb{R}^d$ linear mappings is ε-differentially private (in the l_1 sense).*

Proof. [8] Since the K-norm mechanism is a version of the Laplacian mechanism, it is enough to show that for $q'(x, a) = -||Fx - a||_K$, $\Delta(q') = 1$. Suppose that $||x - y||_1 \leq 1$. Then $\Big|||Fx - a||_K - ||Fy - a||_K\Big| \leq ||F(x - y)||_K \leq 1$, where in the first inequality we use triangle inequality and in the second, $||F(x - y)||_K \leq 1$ since $F(x - y) \in FB_1^n = K$. □

Hardt and Talwar also gives in [8] a method for sampling in polynomial time from the distribution μ_x given by the K-norm mechanism on input x:

1. Sample r from Gamma$(d + 1, 1/\varepsilon)$ distribution, i.e.

$$\mathbf{Pr}[r > R] = \frac{1}{\varepsilon^{-d}\Gamma(d+1)} \int_R^\infty e^{-\varepsilon t} t^d \mathrm{d}t.$$

2. Sample a uniformly from $Fx + rK$.

Since if $||a - Fx||_K = R$, then a can be sampled from $Fx + rK$ for any $r > R$, so the density function of the above distribution is

$$g_x(a) = \frac{1}{\varepsilon^{-d}\Gamma(d+1)} \int_R^\infty \frac{e^{-\varepsilon t} t^d \mathrm{d}t}{\mathrm{Vol}(tK)} =$$

$$= \frac{\int_R^\infty e^{-\varepsilon t} \mathrm{d}t}{\Gamma(d+1)\mathrm{Vol}(\varepsilon^{-1}K)} = \frac{e^{-\varepsilon R}}{\Gamma(d+1)\mathrm{Vol}(\varepsilon^{-1}K)} = f_x(a).$$

Now, we can easily compute the expected error of the K-norm mechanism:

Proposition 2.8. *[8] The K-norm mechanism for $\mathbb{R}^n \to \mathbb{R}^d$ linear mappings has expected l_1 error at most $\frac{d+1}{\varepsilon} E_{z \in K}||z||_1$.*

Proof. [8] Let $\nu =$Gamma$(d+1, \varepsilon^{-1})$.

$$E_{a\sim\mu_x}||Fx-a||_1 = E_{r\sim\nu}E_{z\in rK}||z||_1 = \Big(E_{r\sim\nu}r\Big)\Big(E_{z\in K}||z||_1\Big) =$$

$$= \frac{\Gamma(d+1+1)}{\varepsilon\Gamma(d+1)}E_{z\in K}||z||_1 = \frac{d+1}{\varepsilon}E_{z\in K}||z||_1.$$

□

For $z \in K = FB_1^d$, $||z||_1 \leq \Delta(F) = ||F||$, therefore we have that the expected error of the K-norm mechanism on F is at most $\frac{(d+1)||F||}{\varepsilon}$, which is a $\frac{d+1}{d}$ times worse bound than the $\frac{d||F||}{\varepsilon}$ bound on the error of the exponential (Laplacian) mechanism, but $E_{z\in K}||z||_1$ can be much smaller than $||F||$ for example if $||F||$ is large, but only in "one direction", like if for an orthonormal basis $\{e_1, \ldots, e_n\}$, $||Fe_i||_1 = 1$ for $i = 1, \ldots, n-1$ and $||Fe_n||_1 = c$ where c is some large constant (and $d = n$).

What is the relationship between the K-norm mechanism and the exponential mechanism for linear mappings? The philosophy of the K-norm mechanism is to output any incorrect output with the smallest possible probability regardless of how far they are from the correct output. $||Fx-a||_K$ means how much do we need to perturb the input for getting a instead of Fx as the output of F. For any input x, we want to put weight 1 on Fx, and put the smallest possible weights on the other inputs so as to still achieve ε-differential privacy. If there is an $y \in F^{-1}(a)$ such that $||x-y||_1 = l$, then the possibility of any event at μ_y is at least $\exp(-\varepsilon l)$ times the possibility of the event at μ_x, so we put $\exp(-\varepsilon l)$ on $a = F(y)$. For the exponential mechanism, we have $||Fx - Fy|| \leq ||F|| \cdot ||x-y||$, so $||Fx-a||_K \leq ||Fx-a||/||F||$. But if $||Fx-a||/||F||$ is small, it does not imply that $||Fx-a||_K$ is small. Therefore by the exponential mechanism, we can give some outputs a larger weight than necessary from the closeness of their origins.

In the linear case, the K-norm mechanism is equivalent to the following process: On input $x \in \mathbb{R}^n$, choose an element $y \in x+\mathrm{Ker}(F)^\perp$ with probability proportional to $\exp(-\varepsilon||x-y||_1)$, and then output Fy. This gives the same distribution, because F is injective from $\mathrm{Ker}(F)^\perp$ to $\mathrm{Im}(F)$, and $||Fx-Fy||_K = ||x-y||_1$ if $y \in x+\mathrm{Ker}(F)^\perp$. On $\mathbb{R}^d \setminus \mathrm{Im}(F)$, both mechanism output any element with probability 0. Intuitively, the K-norm mechanism selects a noisy input (amongst the "meaningful" inputs), and then outputs the correct answer corresponding to that input.

As Hardt and Talwar points out, the K-norm mechanism can be extended for the general case as well:

Definition 2.5 (K-norm mechanism)**.** *Define $q'(D,r)$ as* $\min\{k : \exists D' \in \mathcal{D}, d(D,D') = k, F(D') = r\}$*. Then apply the exponential mechanism with q'.*

This definition gives the same notion for the case of linear mappings as Definition 2.4 (naturally we need to interpret $d(D,D')$ as $||D-D'||_1$). Also in this general case, the K-norm mechanism can be interpreted as the mechanism that tries to choose any incorrect output with the smallest possible probability regardless of their distance from the correct output.

Note, that in many cases, the K-norm mechanism coincides with the exponential mechanism. This is the case for example for counting queries, for histograms, and also for the min cut problem.

For being able to prove these statements, note, that if we apply the exponential mechanism with $q(D, r) - \mathrm{OPT}(D)$ instead of $q(D, r)$, than we get the same mechanism, because each term gets multiplied by $\exp(-\mathrm{OPT}(D))$, but the normalization sum also gets multiplied by this value.

For counting queries, if the real output is c, then we can always add t elements that satisfy the property P, and so we can get output $c + t$ with t changes. Similarly if $c > t$, then we can leave t elements that satisfy the property P, and so we can get $c - t$ with t changes.

For histograms, if the real output is $\mathbf{c} \in \mathbb{Z}^d$ and we have $\mathbf{t} \in \mathbb{Z}^d$ such that $||\mathbf{t}||_1 = t$, then $\mathbf{t} = (t_1, \ldots, t_d)$ where $|t_1| + \ldots + |t_d| = t$. Then for each property P_i, we can modify the input on $|t_i|$ elements to get the result $\mathbf{c} + \mathbf{t}$. (This is true because of the disjointness of the properties.) Therefore there exists an input in distance $||\mathbf{t}||_1$, that gives output $\mathbf{c} + \mathbf{t}$.

For the min cut problem, if a min cut in G is C and we have another cut C', then to have C' as a min cut, we can leave $Cost(G, C') - Cost(G, C)$ edges from the cut C'.

3 Some cases when privacy cannot be achieved

There are some cases when ε-differential privacy is not achievable if we require any non-trivial utility. This situation is common if the private input takes its values from an unbounded set (for example $\mathbb{R}$ or $\mathbb{R}^n$) and we measure the distance of the databases by Hamming distance, or if we pose some input-dependent constraint on the output-set of the mechanism (for example, we would like to solve the vertex cover problem, and we require that on any graph, the mechanism should always output a valid vertex cover).

3.1 A sufficient condition

Proposition 3.1. *Let R and $\mathcal{D}$ be discrete sets and $\varepsilon > 0$. If for an $r \in R$ there exists a $D_1 \in \mathcal{D}$ such that $\forall K > 0$ there exists $D_2 \in \mathcal{D}$ such that $d(D_1, D_2) = 1$ and $h(D_2, r) - h(D_1, r) \geq K$, then for each ε-differentially private mechanism M that has error at most a fixed γ, we have $\mathbf{Pr}(M(D) = r) = 0$ for all $D \in \mathcal{D}$.*

Proof. We first prove that $\mathbf{Pr}(M(D_1) = r) = 0$. Then for each $D \in D$ it follows from the ε-differentially privacy of M, that $\mathbf{Pr}(M(D) = r) \leq \exp(\varepsilon(d(D_1, D))) \cdot \mathbf{Pr}(M(D_1) = r) = 0$.

Let us suppose that $\mathbf{Pr}(M(D_1) = r) = p > 0$. There exists a K such that $K \cdot \exp(-\varepsilon) \cdot p > \gamma$. Let us take a D_2 such that $d(D_1, D_2) = 1$ and $h(D_2, r) - h(D_1, r) \geq K$. Then $\mathbf{Pr}(M(D_2) = r) \geq \exp(-\varepsilon)p$. We have

$$E[h(D_2, M(D_2))] \geq h(D_2, r) \cdot \mathbf{Pr}[M(D_2) = r] \geq (h(D_1, r) + K) \cdot \mathbf{Pr}[M(D_2) = r]$$

$$\geq K \cdot \mathbf{Pr}[M(D_2) = r] \geq K \cdot \exp(-\varepsilon) \cdot p >$$

that contradicts the fact that M has expected error at most γ. □

Very similarly, we can give a proposition also for the case when R is not discrete.

Proposition 3.2. *Let R and $\mathcal{D}$ be arbitrary sets, and $\varepsilon > 0$. If for an $S \subseteq R$ there exists a $D_1 \in \mathcal{D}$ such that $\forall K > 0$ $\exists D_2 \in \mathcal{D}$ such that $d(D_1, D_2) = 1$ and $\min\{h(D_2, r) : r \in S\} - \max\{h(D_1, r) : r \in S\} \geq K$, then for each M ε-differentially private mechanism that has error at most a fixed γ, $\mathbf{Pr}(M(D) \in S) = 0$ $\forall D \in \mathcal{D}$.*

Proof. We first prove that $\mathbf{Pr}(M(D_1) \in S) = 0$. Then for each $D \in D$ it follows from the ε-differentially privacy of M, that $\mathbf{Pr}(M(D) \in S) \leq \exp(\varepsilon(d(D_1, D))) \cdot \mathbf{Pr}(M(D_1) \in S) = 0$.

Let us suppose that $\mathbf{Pr}(M(D_1) \in S) = p > 0$. There exists a K such that $K \cdot \exp(-\varepsilon) \cdot p > \gamma$. Let us take a D_2 such that $d(D_1, D_2) = 1$ and $\min\{h(D_2, r) : r \in S\} - \max\{h(D_1, r) : r \in S\} \geq K$. Then $\mathbf{Pr}(M(D_2) \in S) \geq \exp(-\varepsilon)p$. Let f_{d_2} be the density function of the output-distribution of $M(D_2)$. We have

$$E[h(D_2, M(D_2))] = \int_R h(D_2, r) \cdot f_{D_2}(r)dr \geq \int_S h(D_2, r) \cdot f_{D_2}(r)dr \geq$$

$$\geq \int_S (h(D_1, r) + K) \cdot f_{D_2}(r)dr \geq \int_S K \cdot f_{D_2}(r)dr =$$

$$= K \cdot \mathbf{Pr}[M(D_2) \in S] \geq K \cdot \exp(-\varepsilon) \cdot p > \gamma$$

that contradicts the fact that M has expected error at most γ. □

Using these propositions, we can show that some natural problems can only be solved in a differentially private way by adding enormous expected error on some inputs.

Median The (unbounded) median problem is the following: We are given a finite set $D \subset \mathbb{R}$, and we would like to get its median. If $|D|$ is odd, it means the middle element in the ordered version of D. If $|D|$ is even, the median is the average of the two middle elements. An algorithm for the median problem will output a number $r \in \mathbb{R}$. The goodness of the answer will be $q(D, r) = -|median(D) - r|$.

Proposition 3.3. *For any $\varepsilon > 0$ and $\gamma > 0$, there exists no ε-differentially private mechanism, that computes the median with expected error at most γ.*

Proof. We show that such an ε-differentially private algorithm could give an output from any interval $[i, i+1]$, $i \in \mathbb{Z}$ with probability 0. This would be a contradiction if there existed any such algorithm, because in this case $\mathbf{Pr}[M(D) \in \mathbb{R}] \leq \sum_{i \in \mathbb{Z}} \mathbf{Pr}[M(D) \in [i, i+1]] = 0$ for any input D.

We use Proposition 3.2. For an $i \in \mathbb{Z}$, take $D_1 = \{i\}$. For a $K > 0$, take $D_2 = \{i, i+2K+4\}$. Then $d(D_1, D_2) = 1$. As $median(D_1) = i$ and $median(D_2) = i + K + 2$, $\min\{h(D_2, r) : r \in [i, i+1]\} = K + 1$ and $\max\{h(D_1, r) : r \in [i, i+1]\} = 1$. So $\min\{h(D_2, r) : r \in S\} - \max\{h(D_1, r) : r \in S\} = K$ □

From the proof of Proposition 3.2, we can also see, that there exists no ε-differentially private mechanism, that computes the median with expected error at most γ on databases D with at most two elements. Thus, for the median problem, we cannot give an ε-differentially private mechanism with some error bound $err(|D|)$ depending on the cardinality of the input database, like we did for example in the case of the min cut.

This kind of problem can be solved by bounding the range of possible inputs: For example we can consider the following (bounded) median problem: the input databases take their elements from $[a, b]$, where $a, b \in \mathbb{R}$. In the real word applications this is typically a natural assumption. In this case, $\Delta(q) = \frac{a+b}{2}$ is finite, so the exponential mechanism with $\varepsilon' = \varepsilon/(a+b)$ gives an ε-differentially private algorithm. If instead of the exponential mechanism, we use the Laplacian mechanism, i.e. we add noise with Laplace distribution, and if we get a result outside $[a, b]$, then we give a is the result was smaller than a and we give b if it was larger than b, then we get only a bit larger error than as if we concentrate the exponential mechanism on $[a, b]$, but we get expected error $\frac{b-a}{\varepsilon}$ which makes the result practically meaningless. Also in the general case, we can prove that we need an unacceptably high amount of noise to achieve differential privacy even is this bounded case:

Proposition 3.4. *Any ε-differentially private algorithm solving the bounded median problem on range $[a, b]$ must add expected error at least $\frac{b-a}{8}$ on some input for any $\varepsilon < 1/2$.*

Proof. We can assume that $a = 0$, $b = K$, because otherwise the inputs and outputs can be shifted. Take the following three inputs: $D_0 = \{0, 0, K\}$, $D = \{0, K/2, K\}$, $D_K = \{0, K, K\}$. Then the distance between any two of these inputs is 2.

Suppose that there exists an ε-differentially private algorithm M with error less that $K/8$. Then $\mathbf{Pr}[M(D_0) \in [0, K/4]] \geq 1/2$ and $\mathbf{Pr}[M(D_K) \in [3K/4, K]] \geq 1/2$ from Markov-inequality. From the ε-differential privacy of M, $\mathbf{Pr}[M(D) \in [0, K/4]] \geq 1/2 \cdot \exp(-2\varepsilon) \geq 1/2 \cdot \exp(-1)$ and $\mathbf{Pr}[M(D_K) \in [3K/4, K]] \geq 1/2 \cdot \exp(-2\varepsilon) \geq 1/2 \cdot \exp(-1)$.

Thus $\mathbf{Pr}[M(D) \in [K/4, 3K/4]] \leq \exp(-1) < 1/2$, so
$\mathbf{Pr}[M(D) \notin [K/4, 3K/4]] > 1/2$. We conclude that $E[|M(D) - median(D)|] > K/8$ which is a contradiction with our assumption. □

We can see, that for the median problem, there exist some very sensitive inputs, for which a private mechanism needs to add a large error. It can happen though, that there are only a few such inputs, and the "typical" input is not very sensitive. Therefore we can try to give a mechanism that optimizes the amount of noise for each input instead of the worst case, and examine the expected error on individual inputs. In this case the mechanism will give a large expected error in the worst case, but we might be able to prove, that on "interesting" inputs, it gives good accuracy.

A theory that focuses on this model is smoothed sensitivity [11]. This theory considers the case when the output space is $\mathbb{R}^d$, and the distance on databases is measured by Hamming distance. The authors define an input-dependent smooth sensitivity quantity $S^*_F(D)$, and then use it instead of the global sensitivity $\Delta(F) = \max\{||F(D_1) - F(D_2)|| : \ d(D_1, D_2) = 1\}$. According to them, if we take $LS_F(D) = \max\{||F(D) - F(D')|| : \ d(D, D') = 1\}$ and

$$S^*_F(D) = \max\{LS_F(D') \cdot \exp[-(\varepsilon/\beta) d(D, D')] : \ D' \in \mathcal{D}\},$$

where $\beta = \ln(1/\delta) \cdot \sqrt{d}$ if $d > 1$, and $\beta = \ln(1/\delta) \cdot 2$ if $d = 1$, then for $\mathcal{A}(D) = F(D) + Z$, where $Z \sim \mathrm{Laplace(0, S^*_F(D)/\varepsilon)}$, mechanism $\mathcal{A}$ is (ε, δ)-differentially private. In this case we can immediately see, that the mechanism gives expected error $S^*_F(D)/\varepsilon$ on each input D.

For the median problem ($F = \mathit{median}$), if $D = \{x_1, \dots, x_{2m+1}\}$ in increasing order, then $LS_F(D) = \max\{x_{m+1} - x_m, x_m - x_{m-1}\}$, and by changing $k < m$ elements, the sensitivity can grow to at most $\max\{x_{m+k+1} - x_m, x_m - x_{m-k-1}\}$. Therefore on typical inputs, the smooth sensitivity is much smaller than the global (worst case) sensitivity.

We note, that though the Laplacian mechanism gives the same amount of error on each input, this is not always true for the exponential mechanism: in the case of the min cut problem, on the empty graph, the mechanism gives 0 expected error, while this is not true in general. Therefore one could also study for the exponential mechanism how much expected error it gives on an individual input.

The situation is even better for the K-norm mechanism. In fact, the K-norm mechanism also tailors the amount of noise to each individual input. Take for example the bounded median problem. For simplicity, consider the case where the number of elements in the input database is fixed and $d(D_1, D_2) = 1$ if we get D_2 from D_1 by changing an element. If $D = \{x_1, \dots, x_{2m+1}\}$, then the K-norm mechanism gives weight 1 to x_m, weight $\exp(-\varepsilon)$ to the numbers in $[x_{m-1}, x_m)$ and $(x_m, x_{m+1}]$, $\exp(-2\varepsilon)$ to the numbers in $[x_{m-2}, x_{m-1})$ and $(x_{m+1}, x_{m+2}]$ and so on. If for example we have $a = 0$, $b = K$, and a database D, where $|D| = n$, the median is x_m and the elements $x_{m-t+1}, \dots, x_{m+t-1}$ lie in the interval $[x_m - \eta, x_m + \eta]$ for some η; then the K-norm mechanism gives expected error at most $\eta + \exp(-\varepsilon t)K/2$ which can be considerably smaller than $K/8$.

3.2 Vertex cover

There is another scenario when privacy is typically not achievable with any reasonable utility. This is the case when the outputs have an input-dependent constraint.

Consider for example the vertex cover problem [7]. Here we have a graph G with n vertices as input. The private information is the edge set of the graph. We wish to get a vertex cover of minimal size. Denote the size of the minimal vertex cover in graph G by $\tau(G)$. It might be important for the application, that the algorithm should always output a real vertex cover. But as the following proposition shows, if we demand this, we can get no private algorithm with any nontrivial utility:

Proposition 3.5. *[7] For any ε, any ε-differentially private mechanism that solves the vertex cover problem by always outputting a real vertex cover must output a vertex set of size at least $n-1$ with probability 1 on any graph G with n vertices.*

Proof. Suppose that such a ε-differentially private mechanism M outputs a vertex set $X \subseteq V - \{u, v\}$ on a graph $G = (V, E)$ with probability $p > 0$. Then X is a vertex cover of G, so (u, v) is not an edge. Take the graph $G' = (V, E \cup \{(u, v)\})$. G and G' differ in one edge, so $\mathbf{Pr}(M(G') = X) \geq \exp(-\varepsilon) \cdot \mathbf{Pr}(M(G) = X) = p > 0$. But this is a contradiction, since X is not a vertex cover for G' (it does not cover (u, v)), so $\mathbf{Pr}(M(G') = X)$ should be 0. □

Note that this proposition can also be proved from Proposition 3.1 if we consider $q(G, U) = -\infty$ for those sets $U \subseteq V$ that does not form a vertex cover in G. (And we take $q(G, U) = -|U|$ in the other cases.) Then the mechanisms giving expected error at most n are exactly the mechanisms that output a valid vertex cover with probability 1. But for any vertex set $U \subseteq V$ where there exists $u, v \in V \setminus U$, for a graph G where U is a vertex cover, $q(G + (u, v), U) = -\infty$ while $q(G, U)$ is finite, therefore $h(G + (u, v), U) - h(G, U) = \infty$. We can conclude that any mechanism that always outputs a valid vertex cover outputs U with probability 0.

There are two possible solutions to this problem. The first possibility is that we do not demand the mechanism to always output a valid vertex cover, only consider the value of an output set X as $q(G, X) = -|X| - c \cdot \sharp\{\text{not covered edges}\}$ (where c is some constant). But if for the application we inevitably need that the output solution is a vertex cover, we can try to design a mechanism that outputs some encoding of a vertex cover, from which we can construct it very simply.

[7] provides a solution of the second type. The idea is to output an orientation of the edge set of the graph. Then the corresponding vertex cover is the set of vertices that have an incoming edge. This is indeed a vertex cover, because each edge is going to be covered by its endpoint in the orientation.

In practice we cannot trivially output an orientation of the edges if the set of edges is a private input. For this reason, Gupta et al. give a bit more special output: they output a permutation of the vertex set. This defines an orientation of the edges by orienting each edge from the endpoint appearing later in the permutation to the endpoint appearing first. There is no restriction for the permutations that can be output for a given input, so this way they avoid the feasibility constraint problem of the original vertex cover problem. From a permutation, we can build up a vertex cover on-line: We put a vertex in the vertex cover if there is an edge going from it to a vertex not seen yet.

This algorithm will be an example for the case where we convert a non private randomized algorithm to be differentially private by making the choices of the algorithm a bit noisy. We will also point out how this noisy sampling is connected to the exponential mechanism. The algorithm is based on a (non private) randomized 2-approximate algorithm for vertex cover by Pitt [12]. The algorithm of Pitt repeatedly selects a vertex with probability proportional to its uncovered degree until there exists an uncovered edge. The differentially private version of the algorithm will select the consecutive vertices from a noisy version of this distribution. The less number of vertices remains, the more noise we will need to add.

Algorithm 5 Vertex cover, outputting permutations [7]

```
Input: $G = (V, E), \varepsilon$
$V_1 := V$, $E_1 := E$, $list := \emptyset$
for $i = 1, \ldots, n$ do
    let $w_i = 4/\varepsilon \cdot \sqrt{n/(n-i+1)}$
    choose vertex $v \in V_i$ with probability proportional to $d_{E_i}(v) + w_i$
    put $v$ at the end of $list$, $V_{i+1} = V_i - \{v\}$, $E_{i+1} = E_i - (\{v\} \times V_i)$
end for
Output: $list$
```

Proposition 3.6. *[7] Algorithm 5 preserves ε-differential privacy.*

Proof. [7] Let us denote the mechanism realizing Algorithm 5 by M. Consider two graphs $G_A = (V, A)$ and $G_B = (V, B)$ where the edge sets differ in one edge. For any permutation π of the vertices, we need to show, that $\mathbf{Pr}(M(G_A) = \pi) \leq \exp(\varepsilon) \cdot \mathbf{Pr}(M(G_B) = \pi)$. Let us denote the uncovered edges after in the i^{th} step of the mechanism by A_i, B_i respectively. Then

$$\mathbf{Pr}(M(G_A) = \pi) = \prod_{i=1}^{n} \frac{d_{A_i}(\pi_i) + w_i}{2|A_i| + (n-i+1)w_i}$$

Similarly

$$\mathbf{Pr}(M(G_B) = \pi) = \prod_{i=1}^{n} \frac{d_{B_i}(\pi_i) + w_i}{2|B_i| + (n-i+1)w_i}$$

Suppose that A contains an extra edge in comparison with B. If this extra edge became covered in π in the j^{th} step, then in steps $i = 1, \dots j-1$, the edges incident to the chosen vertex are the same in A and B. We always only look at uncovered edges, so after the j^{th} step, the uncovered edges incident to the chosen vertex are also the same in A and B. Therefore, we have $d_{A_i}(\pi_i) = d_{B_i}(\pi_i)$ for $i \neq j$, and $d_{A_j}(\pi_i) = d_{B_j}(\pi_j) + 1$.

Also, until the extra edge becomes covered, $|A_i| = |B_i| + 1$ (for $i \leq j$). For $i > j$, we have $|A_i| = |B_i|$.

Therefore in this case,

$$\frac{\mathbf{Pr}(M(G_A) = \pi)}{\mathbf{Pr}(M(G_B) = \pi)} = \prod_{i=1}^{n} \Big[\frac{d_{A_i}(\pi_i) + w_i}{d_{B_i}(\pi_i) + w_i} \cdot \frac{2|B_i| + (n-i+1)w_i}{2|A_i| + (n-i+1)w_i}\Big] \leq$$

$$\leq \frac{d_{A_j}(\pi_j) + w_j}{d_{B_j}(\pi_j) + w_j} \cdot \prod_{i=1}^{j} \frac{2|B_i| + (n-i+1)w_i}{2|A_i| + (n-i+1)w_i} \leq$$

$$\frac{d_{B_j}(\pi_j) + 1 + w_j}{d_{B_j}(\pi_j) + w_j} \cdot \prod_{i=1}^{j} \frac{2|B_i| + (n-i+1)w_i}{2|B_i| + 2 + (n-i+1)w_i} \leq$$

$$\leq (1 + \frac{1}{w_j}) \cdot \prod_{i=1}^{j} 1 \leq \exp(1/w_j) \leq \exp(\varepsilon)$$

since $1/w_j \leq \varepsilon/4 \leq \varepsilon$.

If B contains an extra edge in comparison with A:

$$\frac{\mathbf{Pr}(M(G_A) = \pi)}{\mathbf{Pr}(M(G_B) = \pi)} = \prod_{i=1}^{n} \Big[\frac{d_{A_i}(\pi_i) + w_i}{d_{B_i}(\pi_i) + w_i} \cdot \frac{2|B_i| + (n-i+1)w_i}{2|A_i| + (n-i+1)w_i}\Big] \leq$$

$$\leq \frac{d_{A_j}(\pi_j) + w_j}{d_{B_j}(\pi_j) + w_j} \cdot \prod_{i=1}^{j} \frac{2|B_i| + (n-i+1)w_i}{2|A_i| + (n-i+1)w_i} \leq$$

$$\leq \frac{d_{A_j}(\pi_j) + w_j}{d_{A_j}(\pi_j) + 1 + w_j} \cdot \prod_{i=1}^{j} \frac{2|A_i| + 2 + (n-i+1)w_i}{2|A_i| + (n-i+1)w_i} \leq$$

$$\leq 1 \cdot \prod_{i=1}^{j} \frac{2 + (n-i+1)w_i}{(n-i+1)w_i} \leq \prod_{i=1}^{j} \Big[1 + \frac{2}{(n-i+1)w_i}\Big] \leq \prod_{i=1}^{j} \exp\Big[\frac{2}{(n-i+1)w_i}\Big] =$$

$$= \exp\Big(\sum_{i=1}^{j} \frac{2}{(n-i+1)w_i}\Big) = \exp\Big(\frac{\varepsilon}{2} \sum_{i=1}^{j} \frac{1}{\sqrt{n(n-i+1)}}\Big) \leq \exp(\varepsilon)$$

since $\frac{\varepsilon}{2} \sum_{i=1}^{j} \frac{1}{\sqrt{n(n-i+1)}} \leq \frac{\varepsilon}{\sqrt{n}} \sum_{i=1}^{n} \frac{1}{2\sqrt{i}} \leq \frac{\varepsilon}{\sqrt{n}} \int_0^n \frac{1}{2\sqrt{x}} dx = \frac{\varepsilon}{\sqrt{n}} \sqrt{n} = \varepsilon.$ □

Proposition 3.7. *[7] For any graph G, the expected size of the vertex cover induced by the permutation output by Algorithm 5 is at most $(2 + \frac{2}{n} \sum_{i=1}^{n} w_i) \cdot \tau(G) \leq (2 + 16/\varepsilon) \cdot \tau(G)$.*

Our proof will be based on [7], but we point out a small error in their proof.

Proof. We will prove a slightly more general result. Consider Algorithm 5 with the choice of w_i as $w_i^n = 4/\varepsilon \cdot \sqrt{N/(n-i+1)}$, where $N \leq n$ is some integer. This corresponds to the original version of Algorithm 5 started for a graph with N vertices, but where we only look at the algorithm from the $N-n+1^{th}$ phase. Let us denote this mechanism by $M_{N,n}$, and the size of the vertex cover induced by the permutation given by $M_{N,n}$ on a graph G on n vertices by $size(M_{N,n}(G))$. We will show, that $E(size(M_{N,n}(G))) \leq (2+\frac{2}{n}\sum_{i=1}^n w_i^n) \cdot \tau(G)$ for any N, ε and graph G on n vertices. And we prove that for $N = n$, $(2+\frac{2}{n}\sum_{i=1}^n w_i^n) \leq (2+16/\varepsilon)$. The proposition is a special case with $N = n$.

We proceed by induction on n.

For any N, if $n = 1$, the claim holds trivially.

For a general n, fix an optimal vertex cover $X \subseteq V$. Let G have m edges. If $\tau(G) \geq n/2$, the claim is true, since the algorithm can give a vertex cover of size at most n. Suppose that $\tau(G) < n/2$.

Let us denote the random vertex output by $M_{N,n}$ in the first phase by v. v will be an element of the induced vertex cover if and only if $d(v) \neq 0$. If we continue the algorithm after choosing v, it will be exactly the same as running $M_{N,n-1}$ on the graph $G-v$. So we have: On each ω corresponding to the choices of $M_{N,n}$:

$$size(M_{N,n}(G))(\omega) = \chi_{\{d(v)\neq 0\}}(\omega) + size(M_{N,n-1}(G-v))(\omega).$$

Therefore

$$E[size(M_{N,n}(G))] = \mathbf{Pr}[d(v) \neq 0] + E_v E[size(M_{N,n-1}(G-v))]. \tag{2}$$

From the inductive hypothesis, $E[size(M_{N,n-1}(G-v))] \leq (2+\frac{2}{n-1}\cdot\sum_{i=1}^{n-1} w_i^{n-1}) = (2+\frac{2}{n-1}\cdot\sum_{i=2}^n w_i^n)\cdot\tau(G-v)$, because the $w_1^{n-1}, \ldots, w_{n-1}^{n-1}$ corresponding to $M_{N,n-1}$ are exactly the $w_2^n, \ldots, w_n^n$ corresponding to $M_{N,n}$.

$$E[size(M_{N,n}(G))] = \mathbf{Pr}[d(v) \neq 0] + (2+\frac{2}{n-1}\cdot\sum_{i=2}^n w_i^n)\cdot E_v[\tau(G-v)] \tag{3}$$

Now, examine the relationship of $\tau(G)$ and $E_v[\tau(G-v))]$.

If we take v from X (the fixed optimal set), the optimum of $G-v$ is exactly one smaller than the optimum of G. Otherwise all we know is that the optimum of $G-v$ is smaller or equal to the optimum of G. Therefore on each ω corresponding to the choices of $M_{N,n}$: $\quad \chi_{\{v\in X\}}(\omega) \leq \tau(G) - \tau(G-v)(\omega)$. Taking expectation:

$$\mathbf{Pr}(v \in X) \leq \tau(G) - E_v[\tau(G-v)]. \tag{4}$$

By a simple computation

$$\mathbf{Pr}(v \in X) \leq \frac{\sum_{u\in X}(d(u)+w_1^n)}{\sum_{u\in V}(d(u)+w_1^n)} \leq \frac{m+|X|\cdot w_1^n}{2m+nw_1^n} = \frac{m+\tau(G)\cdot w_1^n}{2m+nw_1^n}$$

From the fact, that $(a+b)/(c+d) \geq \min\{a/c, b/d\}$, we have

$$\mathbf{Pr}(v \in X) \geq \min\Big\{\frac{OPT(G)}{n}, \frac{1}{2}\Big\} \geq \frac{OPT(G)}{n} \tag{5}$$

since $\tau(G) < n/2$. From (4) and (5),

$$E_v[\tau(G-v)] \leq \tau(G) - \mathbf{Pr}(v \in X) \leq (1 - \frac{1}{n})\tau(G).$$

Now we can bound the elements of (3) using $\tau(G)$.

$\mathbf{Pr}[d(v) \neq 0] \leq \frac{2m+2mw_1^n}{2m+nw_1^n}$, because the number of such vertices is at most $2m$. Therefore

$$\mathbf{Pr}[d(v) \neq 0] \leq (2+2w_1^n) \cdot \frac{m}{2m+nw_1^n} \leq (2+2w_1^n)\ \mathbf{Pr}(v \in X) \leq$$

$$\leq (2+2w_1^n) \cdot \frac{OPT(G)}{n}.$$

Putting all this together,

$$E[size(M_{N,n}(G))] = \mathbf{Pr}[d(v) \neq 0] + (2 + \frac{2}{n-1} \cdot \sum_{i=2}^{n} w_i^n) \cdot E_v[\tau(G-v)] \leq$$

$$\leq (2+2w_1^n) \cdot \frac{OPT(G)}{n} + (2 + \frac{2}{n-1} \cdot \sum_{i=2}^{n} w_i^n) \cdot \frac{n-1}{n}\tau(G) = (2 + \frac{2}{n} \cdot \sum_{i=1}^{n} w_i^n) \cdot \tau(G).$$

It remains to prove, that for the case of $N = n$, $\frac{1}{n}\sum_{i=1}^{n} w_i^n \leq 8/\varepsilon$. But this is true, since

$$\frac{1}{n}\sum_{i=1}^{n} w_i^n = \frac{1}{n}\sum_{i=1}^{n} \frac{4}{\varepsilon}\sqrt{\frac{n}{n-i+1}} = \frac{4}{\varepsilon\sqrt{n}}\sum_{i=1}^{n} \frac{1}{\sqrt{i}} \leq \frac{4}{\varepsilon\sqrt{n}} 2\sqrt{n} = \frac{8}{\varepsilon},$$

where the fact that $\sum_{i=1}^{n} \frac{1}{\sqrt{i}} \leq 2\sqrt{n}$ was shown in the proof of Proposition 3.6.

□

We point out a small error in the proof of Proposition 3.7 given in [7]. Let us denote the mechanism realizing Algorithm 5 for a graph with n vertices by M_n. With our previous notation, this means $M_n = M_{n,n}$. In the proof in [7], they used induction to prove that $E[size(M_n(G))] \leq (2 + \frac{2}{n}\sum_{i=1}^{n} w_i^n) \cdot \tau(G) \leq (2 + 16/\varepsilon) \cdot \tau(G)$. In the inductive step, they claimed, that $E[size(M_n(G))] = \mathbf{Pr}[d(v) \neq 0] + E_v E[size(M_{n-1}(G-v))]$, but this is not true, since if we continue the mechanism M_n from the second step, we do not apply M_{n-1}, but $M_{n,n-1}$ to the graph $G - v$ (the w_i are different). This problem can be solved by proving for each N, that $E(size(M_{N,n}(G))) \leq (2 + \text{avg}_{i=1,\ldots,n} w_i) \cdot \tau(G)$.

The problem of sampling such a permutation is slightly similar to the problem of answering histogram queries in the sense, that we want to output more objects (in the case of the histogram queries: more counts, in this case: more vertices), but the changing of the input changes the q function only for a limited number of outputs. This is the reason why we could manage to achieve in both cases a differentially private algorithm where the privacy parameter does not degrade with the dimension. But in the case of the histogram queries, we output the solutions at the same time, while here, we select the vertices one by one. Therefore the distribution of the output changes slightly also for those vertices, where the degree did not change, because the normalization sum changes. This is why we needed to add more and more noise in the consecutive phases, but even so, the privacy parameter degraded only by a factor of two. (This is why we needed $w_i = 4/\varepsilon \cdot \sqrt{n/(n-i+1)}$ instead of $2/\varepsilon \cdot \sqrt{n/(n-i+1)}$.)

We note that the above idea of sampling vertices with probability proportional to $d(v) + w_i$ can be thought of as an approximate version of the exponential mechanism. In the non private algorithm of Pitt [12], we sample with probability proportional to $d(v)$. Using the exponential mechanism, we would sample with probability proportional to $\exp(\varepsilon d(v))$. In Algorithm 5, in the first phase we choose a vertex v with probability

$$\frac{d(v) + \frac{4}{\varepsilon}}{2m + \frac{4n}{\varepsilon}}$$

Using the exponential mechanism, we would choose v in the first phase with probability

$$\frac{\exp(\varepsilon d(v))}{\sum_{u \in V} \exp(\varepsilon d(u))}$$

Approximating $\exp(x)$ with $1 + x$, we would get

$$\frac{\exp(\varepsilon d(v))}{\sum_{u \in V} \exp(\varepsilon d(u))} \approx \frac{1 + \varepsilon d(v)}{n + \varepsilon \cdot 2m} = \frac{d(v) + \frac{1}{\varepsilon}}{2m + \frac{n}{\varepsilon}}$$

which means, that the first phase of Algorithm 5 can be thought of as an approximate version of the exponential mechanism with $\varepsilon' = \varepsilon/4$. In the subsequent steps, ε' is degraded slightly to maintain the privacy level.

If $q(D, r) > 0$ for each $D \in \mathcal{D}$ and $r \in R$, and $\varepsilon \leq \frac{1}{3\Delta(q)}$, then this "approximate" exponential mechanism of sampling r with probability proportional to $1 + \varepsilon q(D, r)$ on input D is $3\varepsilon\Delta(q)$-differentially private in itself:

Theorem 3.1. *Let $\varepsilon > 0$ and R be finite. If $q(D, r) > 0$ for each $D \in \mathcal{D}$ and $r \in R$, and $\varepsilon \leq \frac{1}{3\Delta(q)}$ then the mechanism giving $r \in R$ with probability proportional to $1 + \varepsilon q(D, r)$ on input D is $3\varepsilon\Delta(q)$-differentially private.*

Proof. Take $D_1, D_2 \in \mathcal{D}$, $d(D_1, D_2) = 1$. For shortness, use the notation $\Delta(q) = \Delta$.Then

$$\frac{1 + \varepsilon q(D_1, r)}{\sum_{s \in R}(1 + \varepsilon q(D_1, s))} \leq \frac{1 + \varepsilon\Delta + \varepsilon q(D_2, r)}{|R| - \varepsilon\Delta|R| + \varepsilon \sum_{s \in R} q(D_2, s)} \leq$$

$$\leq \frac{1+\varepsilon\Delta+\varepsilon q(D_2,r)+\varepsilon^2\Delta q(D,r)}{|R|-\varepsilon\Delta|R|+\varepsilon\sum_{s\in R}q(D_2,s)-\varepsilon^2\Delta\sum_{s\in R}q(D_2,s)} =$$

$$= \frac{(1+\varepsilon\Delta)(1+\varepsilon q(D_2,r))}{(1-\varepsilon\Delta)(|R|+\varepsilon\sum_{s\in R}q(D_2,s))} \leq (1+3\varepsilon\Delta)\frac{1+\varepsilon q(D_2,r)}{|R|+\varepsilon\sum_{s\in R}q(D_2,s)} \leq$$

$$\leq \exp(3\varepsilon\Delta)\frac{1+\varepsilon q(D_2,r)}{|R|+\varepsilon\sum_{s\in R}q(D_2,s)}$$

since $\frac{1+\varepsilon\Delta}{1-\varepsilon\Delta} \leq 1+3\varepsilon\Delta$ if $\varepsilon\Delta \leq 1/3$. □

This approximate exponential mechanism is very simple from the sampling point of view: If $\max\{q(D,r) : r \in R\}/\text{avg}\{q(D,r) : r \in R\}$ is polynomial in the size of the input (which is a typical case), then we can sample from the above distribution in polynomial time using rejection sampling:

We take a rectangle $Q = [0,1] \times [0, 1+\max\{q(D,r) : r \in R\}]$ and map R to $[0,1]$ in a uniform way, then we sample uniformly from Q. If we get (x,y), we check, whether $y \leq 1+q(D,r)$ on the output r corresponding to x. If yes, we output r. If not, we start again. Finally we will have an element sampled from the desired distribution. We are successful in a sampling phase with probability $(1+\text{avg}\{q(D,r) : r \in R\})/(1+\max\{q(D,r) : r \in R\})$. The expected number of phases needed is the reciprocal of this. Therefore if $\max\{q(D,r) : r \in R\}/\text{avg}\{q(D,r) : r \in R\}$ is polynomial in the size of the input, then we can sample in polynomial time.

Unfortunately, in the general case, the expected error of this mechanism would be very poor. This is the reason why it is not used in itself. It is an interesting phenomenon, that it is sufficiently accurate for the vertex cover case.

4 Lower bounds

In this section, we repeat Theorem 4.1 from [8], that gives a lower bound on the error needed to add to get a differentially private mechanism for computing a linear function. We show, that the structure of its proof gives a general framework for proving lower bounds on the error needed to add for achieving ε-differential privacy. To illustrate this, we include lower bound proofs following this scheme for various problems. This means slight modifications of the original proofs. Note that Proposition 3.4 about the lower bound on the error of the bounded median also fits this framework.

Theorem 4.1. *[8] Let $\varepsilon > 0$ and $F : \mathbb{R}^n \to \mathbb{R}^d$ be a linear mapping. Let $K = FB_1^n$. Let $h(x,r) = ||Fx-r||_2$. Then for each ε-differentially private mechanism M, $\text{err}(M,F) \geq \Omega(\varepsilon^{-1}d\sqrt{d}\cdot\text{Vol}(K)^{1/d})$.*

Definition 4.1 (δ-packing)**.** *$Y \in \mathbb{R}^d$ is a δ-packing, if for each $x, y \in Y$, $x \neq y$ we have $||x-y||_2 \geq \delta$.*

Lemma 4.1. *[8] A set $K \in \mathbb{R}^d$ contains a $\Omega(\mathrm{Vol}(K)^{1/d}\sqrt{d})$-packing of size at least* $\exp(d)$.

Proof. For a maximal δ-packing Y in K, if we take the ball of radius δ around each point $y \in Y$ (in norm $||.||_2$), then these balls cover K. Otherwise we could add an uncovered point to Y and would still have a δ-packing.

$\mathrm{Vol}(\delta B_2^d) = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)}\delta^d$

Therefore if we have $\delta = \frac{(\Gamma(\frac{d}{2}+1))^{1/d}}{4\sqrt{\pi}}(\mathrm{Vol}(K))^{1/d}$, then the volume of a ball with radius δ is going to be $\mathrm{Vol}(\delta B_2^d) = \frac{\mathrm{Vol}(K)}{4^d}$.

For covering K with balls of such volume, we need at least 4^d balls, therefore a maximal $\frac{(\Gamma(\frac{d}{2}+1))^{1/d}}{4\sqrt{\pi}}(\mathrm{Vol}(K))^{1/d}$-packing has size at least $\exp(d)$. It is enough to show now, that $(\Gamma(\frac{d}{2}+1))^{1/d} = \Omega(\sqrt{d})$ which means that
$\frac{(\Gamma(\frac{d}{2}+1))^{1/d}}{4\sqrt{\pi}}(\mathrm{Vol}(K))^{1/d} = \Omega(\mathrm{Vol}(K)^{1/d}\sqrt{d})$.

Applying Stirling's formula for the gamma function, we get, that
$\Gamma(\frac{d}{2}+1))^{1/d} \sim \left((\frac{4\pi}{d})^{1/2}(\frac{d}{2e})^{d/2}\right)^{1/d} = (\frac{4\pi}{d})^{1/2d}(\frac{d}{2e})^{1/2}$. Therefore indeed we have $(\Gamma(\frac{d}{2}+1))^{1/d} = \Omega(\sqrt{d})$. □

Proof. Proof of Theorem 4.1 [8]. Let $\lambda \geq 1$. From the Lemma, there exists a $\Omega(\lambda\mathrm{Vol}(K)^{1/d}\sqrt{d})$-packing of size $\exp(d)$ in $\lambda K = \lambda F B_1$. Let $Y \subseteq \lambda K$ be such a packing. Take a set X so that $Y = FX$ and $|X| = |Y|$. From the linearity of F, $\lambda K = \lambda F B_1 = F\lambda B_1$, so we can suppose that $X \subseteq \lambda B_1$, or in other words, that for each $x \in X$, $||x||_1 \leq \lambda$.

Suppose by contradiction, that we are given a ε-differentially private mechanism M where $err(M,F) = cd\sqrt{d}\mathrm{Vol}(K)^{1/d}/\varepsilon$ for some c. For any $x \in \mathbb{R}^n$ us denote the distribution of $M(x)$ by μ_x. Let $\lambda = d/(2\varepsilon)$. From the Markov-inequality we have

$$\mathbf{Pr}(||M(x) - Fx||_2 > t) \leq \frac{cd\sqrt{d}\mathrm{Vol}(K)^{1/d}}{\varepsilon t} = \frac{c2\lambda\sqrt{d}\mathrm{Vol}(K)^{1/d}}{t}.$$

For $t = c4\lambda\sqrt{d}\mathrm{Vol}(K)^{1/d})$ we get

$$\mathbf{Pr}(||M(x) - Fx||_2 > c4\lambda\sqrt{d}\mathrm{Vol}(K)^{1/d}) \leq 1/2.$$

Let ud denote the ball around Fx with radius $c4\lambda\sqrt{d}\mathrm{Vol}(K)^{1/d}$ by B_x. Then $\mu_x(B_x) \geq 1/2$ for each $x \in X$.

Since Y is an $\Omega(\lambda\mathrm{Vol}(K)^{1/d}\sqrt{d})$-packing, for small enough c the balls B_x $x \in X$ are disjoint. Fix such a small c.

Then $\mu_0(B_x) \geq \exp(-\varepsilon\lambda)\mu_x(B_x) \geq \frac{1}{2}\exp(-d/2)$ for each $x \in X$ from the differential privacy of M. Since the B_x balls are disjoint, we get that

$$\mu_0(\mathbb{R}^d) \geq \sum_{x\in X}\mu_0(B_x) \geq \exp(d)\cdot\frac{1}{2}\exp(-d/2) > 1.$$

This is a contradiction. We conclude that any ε-differentially private mechanism has error at least $\Omega(\varepsilon^{-1}d\sqrt{d} \cdot \mathrm{Vol}(K)^{1/d})$ of F. □

Now, let us see a lower bound result for the min cut problem. Remember, that for min cut, $q(G, C) = -Cost(G, C)$, where $Cost(G, C)$ is the size of cut C in graph G. The following result is from [7]. The proof given here is a slightly modified version of the original one, modified in order to resemble the proof of Theorem 4.1.

Theorem 4.2. *[7] Any ε-differentially private mechanism has at least $\Omega(\ln n/\varepsilon)$ additive error on min cut for any $\varepsilon \in (3\ln n/n, 1/12)$ and $n \geq 3$.*

Note that this lower bound is tight as Algorithm 2 achieves this error.

Proof. We use a lemma from [7].

Lemma 4.2. *[7] For $\varepsilon \in (3\ln n/n, 1/12)$ there exists a $\ln n/(3\varepsilon)$-regular graph such that minimal cuts are exactly the cuts separating one node, and all the other cuts have size at least $\ln n/(2\varepsilon)$.*

Suppose that we are given an ε-differentially private mechanism M with error $c \cdot \ln n/\varepsilon$ on min cut for some $c \in \mathbb{R}$.

Let G be a graph discribed in the Lemma.

Take the graphs $G_1, \ldots, G_n$ that we get from G by leaving all the edges incident to node $1, \ldots, n$ respectively.

For each i, G and G_i differs in exactly $\ln n/(3\varepsilon)$ edges. The size of the minimal cut in G_i is 0. The size of all the other cuts is at least $\ln n/(6\varepsilon)$, because for $j \neq i$ we leave at most one edge from the cut $(j, V - j)$ (G is a simple graph), and for $\varepsilon < 1/12$, $\ln n/(3\varepsilon) - 1 \geq \ln n/(6\varepsilon)$. The size of the other cuts was at least $\ln n/(2\varepsilon)$, so leaving $\ln n/(3\varepsilon)$ edges, at least $\ln n/(6\varepsilon)$ edges remain.

From the Markov-inequality

$$\mathbf{Pr}(-q(x, M(x)) \geq t) = \mathbf{Pr}(\mathrm{OPT}(x) - q(x, M(x)) \geq t)$$

$$\leq \frac{E(\mathrm{OPT}(x) - q(x, M(x)))}{t} = \frac{\mathrm{err}(M, F)}{t} \leq \frac{c \cdot \ln n}{\varepsilon t}$$

Therefore for $c = 1/12, t = \ln n/6\varepsilon$

$$\mathbf{Pr}(M(G_i) \neq \{i\}) = \mathbf{Pr}(-q(x, M(x)) \geq t) \leq \frac{1}{2}.$$

Or in other words

$$\mathbf{Pr}(M(G_i) = \{i\}) \geq \frac{1}{2}.$$

Since G and G_i differs in $\ln n/(3\varepsilon)$ edges,

$$\mathbf{Pr}(M(G) = \{i\}) \geq \exp(-\ln n/3)\frac{1}{2} \geq \frac{1}{2n^{1/3}}.$$

$$1 \geq \sum_{i=1}^{n} \mathbf{Pr}(M(G) = \{i\}) \geq n \cdot \frac{1}{2n^{1/3}} > 1$$

if $n \geq 3$. this is a contradiction. Therefore all ε-differentially private mechanisms have error $\Omega(\ln n/\varepsilon)$ on min cut. □

One can also give a lower bound for the vertex cover problem, in the case where we output permutations of the vertex set. The following result is a slightly weaker version of the result of [7]. There, they prove lower bounds for the class of mechanism solving vertex cover by outputting an orientation of the edges, while we give a theorem about the class of mechanisms outputting permutations of the vertex set. Here also, the proof uses the construction of [7].

Theorem 4.3. *[7] Any ε-differentially private mechanism that outputs a permutation of the vertices of the input graph gives $\Omega(1/\varepsilon)$ multiplicative error on the size of the induced vertex cover for any $\varepsilon \in (\frac{1}{n}, 1]$.*

Proof. First, let $n = \lceil 1/\varepsilon \rceil$, and $V = \{1, 2, \ldots, \lceil 1/\varepsilon \rceil\}$.

Let G be the graph on V with no edges, and let G_i be the star rooted at i for each $i \in V$. ($G_i = (V, E_i)$, where $E_i = \{(i, j) : j \neq i\}$). Then the optimal vertex cover in G_i is $\{i\}$, so $\mathrm{OPT}(G_i) = 1$ for each $i = 1, \ldots, \lceil 1/\varepsilon \rceil$.

Let us suppose that we have an ε-differentially private algorithm M, that outputs permutations, and that induces a vertex cover with expected size $\frac{c}{\varepsilon} \cdot \mathrm{OPT}(G)$ for each graph G for some fixed constant c. Then on each G_i, it induces a vertex cover of expected size $\frac{c}{\varepsilon}$.

Let us denote the output-distribution of $M(G_i)$ (on the permutations of V) by μ_i, and the output-distribution of $M(G)$ by μ_0. Let us again use the notation $size(M(G))$ for the size of the vertex cover induced by the permutation output by M (which is a random variable here).

From the Markov-inequality,

$$\mathbf{Pr}(size(M(G)) > t) \leq \frac{E[size(M(G))]}{t} \leq \frac{c}{\varepsilon t}.$$

For a permutation π of the vertices, where i is the t^{th} in the order ($i = \pi(t)$), the induced vertex cover in G_i has t vertices: $\{\pi(1), \ldots, \pi(t)\}$. Let us denote by $S_{i,t}$ the set of permutations of V, where i is at most the t^{th} ($\pi^{-1}(i) \leq t$). Then we have: $\mu_i(S_{i,t}) > 1 - \frac{c}{\varepsilon t}$ for each $t \in \{1, 2, \ldots, \lceil 1/\varepsilon \rceil\}$ and for each $i \in V$.

For each i, G and G_i differs in $\lceil 1/\varepsilon \rceil - 1 \leq 1/\varepsilon$ edges. Therefore from the ε-differential privacy of M, $\mu_0(S_{i,t}) \geq \exp(-\varepsilon \cdot \frac{1}{\varepsilon})\mu_i(S_{i,t}) > \exp(-1)(1 - \frac{c}{\varepsilon t})$.

The sets $S_{i,t}$ for $i = 1, \ldots n$ cover the set of permutations of V exactly t times, because any permutation π occurs exactly in $S_{\pi(1),t}, S_{\pi(1),t}, \ldots, S_{\pi(t),t}$.

Therefore we have

$$t \geq \sum_{i=1}^{n} \mu_0(S_{i,t}) \geq n \cdot \exp(-1)(1 - \frac{c}{\varepsilon t}) \geq \frac{1}{\varepsilon} \cdot \exp(-1)(1 - \frac{c}{\varepsilon t}).$$

Written differently

$$t-\frac{1}{\varepsilon}\cdot\exp(-1)+\frac{1}{\varepsilon}\cdot\exp(-1)\frac{c}{\varepsilon t}\geq 0.$$

$$\varepsilon^2 t^2-\exp(-1)\cdot\varepsilon t+\exp(-1)c\geq 0.$$

$$(\varepsilon t-\frac{1}{2e})^2+\frac{c}{e}-\frac{1}{4e^2}\geq 0$$

Fix a t. If c is small enough, the above inequality does not hold, therefore we cannot have such a mechanism for small enough $c=: c_0$.

For a general n, for any ε-differentially private mechanism M, it is enough to show a graph H on $V=\{1,\dots,n\}$ such that $E[size(M(H))]\geq\frac{c_0}{\varepsilon}\mathrm{OPT}(H)$. Let $l=\lfloor\frac{n}{\lceil 1/\varepsilon\rceil}\rfloor$. Partition $V=\{1,\dots,n\}$ to sets $V_0,V_1,\dots V_l$, where $V_1,\dots V_l$ have size $\lceil 1/\varepsilon\rceil$ each, and V_0 may be empty. Let $H|_{V_i}$ be a star for each i, and let $H|_{V_0}$ be a graph with no edges.

Then from the fact that each V_i is a different connected component,

$$E(size(M(H)))=E\Big[\sum_{i=1}^{l}size(M(H|_{V_i}))\Big]=\sum_{i=0}^{l}E[size(M(H|_{V_i}))].$$

Take the following mechanism M_i: For a graph G on V_i, take the union of the graph G with $H|_{V\setminus V_i}$. Run M on this graph, and take from the output only the permutation corresponding to V_i. Then this is an ε-differentially private mechanism. From the first part of the proof, we know, that each ε-differentially private mechanism gives large error on stars: there is a c_0 such that $E(size(M_i(H|_{V_i})))\geq\frac{c_0}{\varepsilon}\mathrm{OPT}(H|_{V_i})$. Therefore we have

$$E[size(M(H))]=\sum_{i=0}^{l}E[size(M(H|_{V_i}))]=0+\sum_{i=1}^{l}E[size(M_i(H|_{V_i}))]\geq$$

$$\sum_{i=1}^{l}\frac{c_0}{\varepsilon}\cdot\mathrm{OPT}(H|_{V_i})=\frac{c_0}{\varepsilon}\sum_{i=1}^{l}\mathrm{OPT}(H|_{V_i})=\frac{c_0}{\varepsilon}\cdot\mathrm{OPT}(H).$$

□

References

[1] Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristics for k-median and facility location problems. *SIAM J. Comput.*, 33(3):544–562, March 2004.

[2] Aditya Bhaskara, Daniel Dadush, Ravishankar Krishnaswamy, and Kunal Talwar. Unconditional differentially private mechanisms for linear queries. In *Proceedings of the 44th symposium on Theory of Computing*, STOC '12, pages 1269–1284, New York, NY, USA, 2012. ACM.

[3] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer Berlin Heidelberg, 2006.

[4] Cynthia Dwork. Differential privacy: a survey of results. In *Proceedings of the 5th international conference on Theory and applications of models of computation*, TAMC'08, pages 1–19, Berlin, Heidelberg, 2008. Springer-Verlag.

[5] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: privacy via distributed noise generation. In *Proceedings of the 24th annual international conference on The Theory and Applications of Cryptographic Techniques*, EUROCRYPT'06, pages 486–503, Berlin, Heidelberg, 2006. Springer-Verlag.

[6] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third conference on Theory of Cryptography*, TCC'06, pages 265–284, Berlin, Heidelberg, 2006. Springer-Verlag.

[7] Anupam Gupta, Katrina Ligett, Frank McSherry, Aaron Roth, and Kunal Talwar. Differentially private combinatorial optimization. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 1106–1125, Philadelphia, PA, USA, 2010. Society for Industrial and Applied Mathematics.

[8] Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *Proceedings of the 42nd ACM symposium on Theory of computing*, STOC '10, pages 705–714, New York, NY, USA, 2010. ACM.

[9] David R. Karger. Global min-cuts in rnc, and other ramifications of a simple min-out algorithm. In *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, SODA '93, pages 21–30, Philadelphia, PA, USA, 1993. Society for Industrial and Applied Mathematics.

[10] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, pages 94–103, Washington, DC, USA, 2007. IEEE Computer Society.

[11] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, STOC '07, pages 75–84, New York, NY, USA, 2007. ACM.

[12] L. Pitt. A simple probabilistic approximation algorithm for vertex cover. Technical report, Yale University, 1985.