Artificial Intelligence 174 (2010) 551–569

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Optimal query complexity bounds for finding graphs

Sung-Soon Choi [a,1], Jeong Han Kim [a,b,*,2]

[a] Department of Mathematics, Yonsei University, Seoul, 120-749, Republic of Korea
[b] National Institute for Mathematical Sciences, Daejeon, 305-340, Republic of Korea

ARTICLE INFO

Article history:
Received 6 August 2008
Received in revised form 12 February 2010
Accepted 13 February 2010
Available online 21 February 2010

Keywords:
Combinatorial search
Combinatorial group testing
Graph finding
Coin weighing
Fourier coefficient
Pseudo-Boolean function
Littlewood–Offord theorem

ABSTRACT

We consider the problem of finding an unknown graph by using queries with an additive property. This problem was partially motivated by DNA shotgun sequencing and linkage discovery problems of artificial intelligence.

Given a graph, an additive query asks the number of edges in a set of vertices while a cross-additive query asks the number of edges crossing between two disjoint sets of vertices. The queries ask the sum of weights for weighted graphs.

For a graph G with n vertices and at most m edges, we prove that there exists an algorithm to find the edges of G using $\mathcal{O}(\frac{m \log \frac{n^2}{m}}{\log(m+1)})$ queries of both types for all m. The bound is best possible up to a constant factor. For a weighted graph with a mild condition on weights, it is shown that $\mathcal{O}(\frac{m \log n}{\log m})$ queries are enough provided $m \geqslant (\log n)^{\alpha}$ for a sufficiently large constant α, which is best possible up to a constant factor if $m \leqslant n^{2-\varepsilon}$ for any constant $\varepsilon > 0$.

This settles, in particular, a conjecture of Grebinski [V. Grebinski, On the power of additive combinatorial search model, in: Proceedings of the 4th Annual International Conference on Computing and Combinatorics (COCOON 1998), Taipei, Taiwan, 1998, pp. 194–203] for finding an unweighted graph using additive queries. We also consider the problem of finding the Fourier coefficients of a certain class of pseudo-Boolean functions as well as a similar coin weighing problem.

1. Introduction

1.1. Graph finding problem

The problem of finding a graph is stated as follows. Suppose that a graph G has n vertices and at most m edges and that the edges of G are unknown. We may consider two types of queries, *additive queries* and *cross-additive queries*. An additive query asks the number of edges in a set of vertices while a cross-additive query asks the number of edges crossing between two disjoint sets of vertices. The problem is to find the edges of G by using as few queries as possible.

Additive queries have been motivated by a main process in shotgun sequencing [6,20]. Shotgun sequencing is a method to determine the whole genome sequence in an organism's DNA. In shotgun sequencing, it is required to order decoded

* Corresponding author at: National Institute for Mathematical Sciences, Daejeon, 305-340, Republic of Korea.
E-mail addresses: ss.choi@yonsei.ac.kr (S.-S. Choi), jehkim@nims.re.kr (J.H. Kim).

[1] This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (CRI, No. 2008-0054850).
[2] This work was partially supported by Yonsei University Research Funds 2006-1-0078 and 2007-1-0025, and by the second stage of the Brain Korea 21 Project in 2007, and by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2006-312-C00455).

0004-3702/$ – see front matter
doi:10.1016/j.artint.2010.02.003

fragments (called *contigs*) of the genome sequence. Given a set of contigs, a method called the multiplex PCR method [39] tells how many pairs of the contigs are adjacent in the original sequence. Thus, the task of ordering contigs is reduced to the problem of finding an unknown graph, which is a Hamiltonian cycle or path, by using additive queries. See e.g., [20].

Cross-additive queries have been motivated by the problem of finding the Fourier coefficients for a certain class of *pseudo-Boolean* functions. A *pseudo-Boolean* function is a real-valued function defined on the set of binary sequences. It is *k-bounded* if it can be expressed as a sum of subfunctions each of which depends on at most k input bits. For example, given a 2-SAT formula, the number of clauses an assignment satisfies is a 2-bounded pseudo-Boolean function. In molecular biology and biophysics, k-bounded functions have been used to study the evolution of a population of organisms in an environment [25]. Specifically, k-bounded functions with small k have received attention in modeling living systems [26] and real biological objects [17]. In evolutionary computation, k-bounded functions have been also used as a benchmark for comparing heuristic algorithms [18,32,33,38]. Cross-additive queries are used to find the Fourier coefficients of 2-bounded functions. More generally, cross-additive queries for k-bounded hypergraphs can be used to find the Fourier coefficients of k-bounded functions, where k-bounded hypergraphs are hypergraphs whose hyperedges are of size at most k.

An algorithm to find an unknown graph is called *non-adaptive* if each query in the algorithm is independent of the answers for the previous queries. Otherwise, it is called *adaptive*. Non-adaptive algorithms are preferable to adaptive ones particularly when the number of required queries is fairly large and parallel computation is available. There have been a number of papers addressing the problem of finding a graph using additive queries. When the (unknown) graph is a Hamiltonian cycle on n vertices, Grebinski and Kucherov [20] presented an adaptive algorithm using $\mathcal{O}(n)$ additive queries, which is the best possible up to a constant factor. Later, Grebinski and Kucherov [21] provided an extensive work for several types of graphs. In particular, for graphs with maximum degree bounded by d, they proved the existence of a non-adaptive algorithm using $\mathcal{O}(dn)$ additive queries. When the graph is k-degenerate, the existence of a non-adaptive algorithm using $\mathcal{O}(kn)$ additive queries was shown by Grebinski [19].

The fully general case that the graph has n vertices and at most m edges has been a matter of primary concern. It has been conjectured by Grebinski [19] that there exists an algorithm to find the unknown graph using $\mathcal{O}(m)$ additive queries provided that $m = \Omega(n)$. The conjecture has not been settled for a decade. For general m, an adaptive algorithm using $\mathcal{O}(m \log n)$ additive queries by Angluin and Chen [4] is the best known to date. In fact, their algorithm uses less powerful queries called *membership queries*, which ask the oracle only about the existence of an edge in a set of vertices. (There have also been a number of papers addressing the problem of finding a graph using membership queries [2,3,5–7].) Recently, Reyzin and Srivastava [34] presented a simpler adaptive algorithm using $\mathcal{O}(m \log n)$ additive queries. In this paper, we prove the conjecture of Grebinski in a stronger form, namely the existence of a non-adaptive algorithm using $\mathcal{O}(\frac{m \log \frac{n^2}{m}}{\log(m+1)})$ additive queries. This bound is best possible and better than $\mathcal{O}(m)$ if $\log \frac{n^2}{m} \ll \log m$, in particular, $m > \frac{n^2}{\log n}$.

We shall focus on bounds for the number of required cross-additive queries. Note that cross-additive queries are less strong than additive queries since a cross-additive query for the disjoint sets S, T of vertices can be answered by the three additive queries for $S \cup T$, S, and T. It can be easily shown that the converse is not true: For example, for a graph with exactly one edge, $\Omega(\log n)$ cross-additive queries are required to verify that it has only one edge while one additive query is enough to verify the same. In the rest of this paper, we state results with respect to cross-additive queries. The same statements hold for additive queries after simple modifications if necessary.

In this paper, we consider two versions of the graph finding problem. The first one is

Problem 1 *(Unweighted graphs).*

INPUT: an unweighted graph G for which the only information given is that
– G has the vertex set $\{1, \ldots, n\}$ and at most m edges
OUTPUT: the edges of G

For the query complexity of the problem, we have the following.

Theorem 1.1. *There is a non-adaptive algorithm that solves Problem* 1 *using* $\mathcal{O}(\frac{m \log \frac{n^2}{m}}{\log(m+1)})$ *cross-additive queries.*

The second is a generalized one for weighted graphs with a moderate condition on weights of edges.

Problem 2 *(Weighted graphs).*

INPUT: a weighted graph G for which the only information given is that
– G has the vertex set $\{1, \ldots, n\}$ and at most m edges
– the weights of edges of G are between n^{-a} and n^b in absolute value
OUTPUT: the edges of G

For weighted graphs, a cross-additive query asks the sum of weights of the edges crossing between two disjoint sets of vertices. We obtain bounds for the number of cross-additive queries required to solve the problem.

Theorem 1.2. *For any fixed constants $a, b > 0$, there are a constant $\alpha > 0$ and a non-adaptive algorithm that solves Problem 2 using $\mathcal{O}(\frac{m \log n}{\log m})$ cross-additive queries provided $m \geqslant (\log n)^{\alpha}$.*

This extends the result of the conference version [13] that gives the bound when m is at least a constant power of n.[3] Concerning the condition on weights, a remark is provided in Section 7.

Notice that we focused only on query complexity of algorithms. The presented algorithms are not computationally efficient and we do not try to optimize them in time complexity. In terms of query complexity, the bounds in the above theorems are optimal up to a constant factor for all or almost all m by the information-theoretic lower bounds. For unweighted graphs, the bound is optimal for all m. For weighted graphs, the bound is optimal if $m \leqslant n^{2-\varepsilon}$ for any constant $\varepsilon > 0$. We do not know yet that the bound is optimal for m in the other range, and we conjecture that it is so up to a constant factor. Theorem 1.2 implies the existence of an algorithm that finds the weights of all edges with the same query bound. This is because, once all edges are discovered, the weights of edges can be found by using at most m additional queries (one for each edge). It is not difficult to show that the algorithm is optimal by an information-theoretic lower bound argument.

1.2. Related problems

Related to the graph finding problem, we consider the coin weighing problem and the problem of finding the Fourier coefficients of a 2-bounded function.

The problem of finding a graph can be regarded as an extension of the coin weighing problem. In the coin weighing problem, we are given n coins among which there are some counterfeits. All the authentic coins have the same weight and the weight is known. Authentic and counterfeit coins are indistinguishable from each other except that their weights are different. Given a group of coins, the scale tells the sum of weights of the coins. The problem is to find the counterfeits using the scale as few times as possible. There are a few papers addressing the case where the weights of counterfeits are identical [1,29,37]. When the weight of each counterfeit is a positive integer and the total weight of counterfeits is bounded above by m, Lindström [28] constructed a non-adaptive algorithm to find the counterfeits with $\mathcal{O}(m \log n)$ weighings. Under the same condition, Grebinski and Kucherov [21] proved the existence of a non-adaptive algorithm with an optimal number of weighings up to a constant factor. In particular, the algorithm uses $\mathcal{O}(\frac{m \log n}{\log m})$ weighings provided that $m \leqslant n^{1-\varepsilon}$ for any constant $\varepsilon > 0$. In this paper, we show that the bound holds even for the general case that the weight of each counterfeit is nearly arbitrary provided that the number of counterfeits is at most m. The formal definition of the problem concerned is as follows.

Problem 3 *(Coin weighing).*

INPUT: coins for which the only information given is that
- the total number of coins is n, and there are at most m counterfeits among them
- all the authentic coins have the same weight and the weight is known
- the weight difference between each counterfeit and an authentic coin is between n^{-a} and n^{b}

OUTPUT: the set of counterfeit coins

For this problem, we have

Theorem 1.3. *For any fixed constants $a, b > 0$, there is a non-adaptive algorithm that solves Problem 3 with $\mathcal{O}(\frac{m \log n}{\log(m+1)})$ weighings.*

There have been many studies of the problem of finding the Fourier coefficients of a k-bounded pseudo-Boolean function $f : \{0,1\}^n \to \mathbb{R}$ where k is a constant. Kargupta and Park [24] presented a deterministic adaptive algorithm that uses $\mathcal{O}(n^k)$ function evaluations. Later, Heckendorn and Wright [22] proposed a randomized adaptive algorithm. For the k-bounded functions with $\mathcal{O}(n)$ non-zero Fourier coefficients generated from a random model, they analyzed the algorithm to show that, with negligible error probability, it finds the Fourier coefficients in $\mathcal{O}(n^2 \log n)$ function evaluations on average. By analyzing the algorithm of Heckendorn and Wright, Choi, Jung, and Moon [12] proved that, for a k-bounded function with m non-zero Fourier coefficients, $\mathcal{O}(\gamma(n,m) m \log n)$ function evaluations are enough with negligible error probability, where $\gamma(n,m)$ is between $n^{\frac{1}{2}}$ and n depending on m. Recently, Choi, Jung, and Kim [11] provided a randomized adaptive algorithm to find the Fourier coefficients with high probability in $\mathcal{O}(m \log n)$ function evaluations.

---

[3] While this paper was in the review process, we learned that for additive queries, Bshouty and Mazzawi [8] improved our result to get the bound for arbitrary m.

Table 1
Summary of results.

	Coin weighing	Unweighted graphs	Weighted graphs	Fourier coefficients
n	# of coins	# of vertices	# of vertices	# of input variables
m	upper bound on # of counterfeits	upper bound on # of edges	upper bound on # of edges	upper bound on # of non-zero coefficients
range of m	$1 \leqslant m \leqslant n$	$1 \leqslant m \leqslant \binom{n}{2}$	$(\log n)^\alpha \leqslant m \leqslant \binom{n}{2}$	$(\log n)^\alpha \leqslant m \leqslant \binom{n}{2} + n + 1$
condition on weights	between n^{-a} and n^b (weight differences)	N/A	between n^{-a} and n^b in absolute value (edge weights)	between n^{-a} and n^b in absolute value (non-zero coefficients)
query complexity	$\mathcal{O}(\frac{m \log n}{\log(m+1)})$	$\mathcal{O}(\frac{m \log \frac{n^2}{m}}{\log(m+1)})$	$\mathcal{O}(\frac{m \log n}{\log m})$	$\mathcal{O}(\frac{m \log n}{\log m})$
optimality	for $m \leqslant n^{1-\varepsilon}$	for all m	for $m \leqslant n^{2-\varepsilon}$	for all m

For the problem of finding weights as well, all above bounds are optimal for all m.

In this paper, we consider the problem for 2-bounded functions with a mild condition on Fourier coefficients as follows.

Problem 4 *(Fourier coefficients).*

INPUT*:* *a* function f for which the only information given is that
- f is a 2-bounded pseudo-Boolean function defined on $\{0,1\}^n$
- f has at most m non-zero Fourier coefficients
- the non-zero Fourier coefficients of f are between n^{-a} and n^b in absolute value

OUTPUT: the Fourier coefficients of f

As we will see, the problem of finding the Fourier coefficients of a 2-bounded function is reduced to a combination of the coin weighing problem and the graph finding problem using cross-additive queries. We have the following corollary from Theorems 1.2 and 1.3. An algorithm is called *k-round* if the sequence of queries $(q_1, \ldots, q_\ell)$ used by the algorithm can be divided into k parts $(q_1, \ldots, q_{\ell_1}), (q_{\ell_1+1}, \ldots, q_{\ell_2}), \ldots, (q_{\ell_{k-1}+1}, \ldots, q_\ell)$ so that each part is non-adaptive given the previous parts. A non-adaptive algorithm is a one-round algorithm. The k-round algorithms with smaller k are preferable as they can be executed using k sets of parallel computations.

Corollary 1.4. *For any fixed constants $a, b > 0$, there are a constant $\alpha > 0$ and a 4-round algorithm that solves Problem 4 in $\mathcal{O}(\frac{m \log n}{\log m})$ function evaluations provided $m \geqslant (\log n)^\alpha$.*

For the 2-bounded functions described in the corollary, the algorithm improves the bound of Choi, Jung, and Kim [11] by a factor of $\log m$ and it is deterministic and fairly non-adaptive. There have been many papers addressing the problem of finding the Fourier coefficients of Boolean functions [9,23,31]. We, however, think that the problem is not closely related to our problem as it is believed that there are not many non-trivial 2-bounded Boolean functions.

As for the theorems for the graph finding problem, we may show the optimality up to a constant factor for the above two theorems by information-theoretic lower bound arguments. Theorem 1.3 is optimal if $m \leqslant n^{1-\varepsilon}$ for any constant $\varepsilon > 0$. However, for the problem of finding weights of coins, it implies the existence of an algorithm with optimal query complexity for all m. By the same argument, Corollary 1.4 is optimal for all m in the specified range.

Table 1 summarizes our results.

1.3. Organization

To prove our results, we use an inequality describing the anti-concentration property of a sum of independent random variables. It is a generalized Littlewood–Offord theorem [14,30] and easily follows from the previous generalizations including ones in [15,16,27,35,36]. We describe the theorem in the next section. In Section 3, we prove Theorem 1.3 (coin weighing) using probabilistic methods. The proof is relatively easy to follow and it would be a good illustration of how the generalized Littlewood–Offord theorem is used. We prove Theorem 1.1 (unweighted graphs) in Section 4 and Theorem 1.2 (weighted graphs) in Section 5. In Section 6, Corollary 1.4 is proved by showing how the problem of finding the Fourier coefficients of a 2-bounded pseudo-Boolean function is reduced to the graph finding problem and the coin weighing problem. Concluding remarks will follow in the last section.

2. Anti-concentration inequality

Random variables of a certain type are highly concentrated near their means with high probability. There are a few inequalities describing the concentration property. An example is the Chernoff bound [10], which shows that the sum of i.i.d. random variables is highly concentrated near its mean. On the other hand, the Littlewood–Offord theorem [14,30] describes the anti-concentration property of the sum of independent random variables. It gives an upper bound for the probability that the sum of independent random variables having two different values is in an interval. There are several generalizations of the theorem including those in [15,16,27,35,36]. We are going to use the following form, which can be found in [27]. It is actually not hard to derive it using the original Littlewood–Offord theorem.

Theorem 2.1 *(Generalized Littlewood–Offord theorem). Let $a_1, \dots, a_n, b_1, \dots, b_n$ and $s > 0$ be real numbers such that $b_i - a_i \geqslant s$ for $1 \leqslant i \leqslant n$ and let $X_1, \dots, X_n$ be independent random variables. Suppose that there is p with $0 < p < 1$ such that $\Pr[X_i \leqslant a_i] \geqslant p$ and $\Pr[X_i \geqslant b_i] \geqslant p$ for all $1 \leqslant i \leqslant n$. Then, for any real number r,*

$$\Pr\left[r \leqslant \sum_{i=1}^{n} X_i \leqslant r + s\right] \leqslant \frac{2.6}{\sqrt{pn}}.$$

A version we will frequently use is

Corollary 2.2. *Let ρ be a positive integer and s be a positive real number. Let $X_1, \dots, X_n$ be independent random variables at least t X_i's of which satisfy $\Pr[X_i = 0] \geqslant p$ and $\Pr[|X_i| \geqslant s] \geqslant p$ for some p with $0 < p < 1$. Then, the probability that $\sum_{i=1}^{n} X_i$ is in an interval of length ρs is at most $\frac{3.7\rho}{\sqrt{pt}}$.*

Proof. Without loss of generality, it may be assumed that $\Pr[X_i = 0] \geqslant p$ and $\Pr[|X_i| \geqslant s] \geqslant p$ for $i = 1, \dots, t$. This implies that $\Pr[X_i = 0] \geqslant \frac{p}{2}$ and, either $\Pr[X_i \geqslant s] \geqslant \frac{p}{2}$ or $\Pr[X_i \leqslant -s] \geqslant \frac{p}{2}$ for $i = 1, \dots, t$. Since X_i's are independent, by Theorem 2.1, the probability that $\sum_{i=1}^{n} X_i$ conditioned on X_i's with $i > t$ is in an interval of length s is at most $\frac{2.6\sqrt{2}}{\sqrt{pt}} \leqslant \frac{3.7}{\sqrt{pt}}$. Thus, so is the probability that $\sum_{i=1}^{n} X_i$ is in an interval of length s and the corollary follows by union bound. □

3. Coin weighing problem

In this section, we prove Theorem 1.3. As the weight of an authentic coin is known in the coin weighing problem, we may assume that the weight of an authentic coin is 0 and the weight of each counterfeit is non-zero (possibly negative). An instance of the coin weighing problem may be regarded as an n-dimensional vector whose coordinates are given by the weights of coins. For an instance $x = (x_i)_{i=1}^{n}$, let $\mathrm{sp}(x)$ denote the support set of x, that is, the set of counterfeit coins in x. Similarly, a query can be regarded as a binary vector and a sequence of ℓ queries as an $\ell \times n$ binary matrix.

The basic idea for the proof of Theorem 1.3 is as follows. A sequence of queries, or a binary matrix M, *separates* a pair of instances x, y if a query in the sequence yields different answers for the two instances, i.e., $Mx \neq My$. A sequence of queries *separates* a set of instances if it separates every pair of instances in the set with distinct supports. In a non-adaptive algorithm, its sequence of queries must separate the set of all instances. If there were only finitely many possible instances, such a sequence might be enough as the given instance can be identified by checking all possible instances. Since there are uncountably many instances, we consider a set of discretized instances: Let $\mathcal{I}$ be the set of all instances $x = (x_i)$ such that x_i is a multiple of $n^{-(a+3)}$. We prove the existence of a sequence $\mathcal{S}$ of queries separating $\mathcal{I}$ by more than $\frac{m}{n^{a+3}}$, i.e., $\|Mx - My\|_\infty > \frac{m}{n^{a+3}}$ for the binary matrix M induced by $\mathcal{S}$ and $x, y \in \mathcal{I}$ with distinct supports. We will write $M_{\mathcal{S}}(x)$ for Mx when we need to specify the sequence $\mathcal{S}$.

Lemma 3.1. *There exists a sequence $\mathcal{S}$ of ℓ queries with $\ell = \mathcal{O}(\frac{m \log n}{\log(m+1)})$ such that*

$$\left\|M_{\mathcal{S}}(x) - M_{\mathcal{S}}(y)\right\|_\infty > \frac{m}{n^{a+3}}$$

for all $x, y \in \mathcal{I}$ with $\mathrm{sp}(x) \neq \mathrm{sp}(y)$.

Once such a sequence $\mathcal{S}$ of queries is found, the following holds.

Lemma 3.2. *Suppose that x is a given instance. Then,*

(i) *there exists $y \in \mathcal{I}$ such that $\mathrm{sp}(x) = \mathrm{sp}(y)$ and*

$$\left\|M_{\mathcal{S}}(x) - M_{\mathcal{S}}(y)\right\|_\infty \leqslant \frac{m}{2n^{a+3}};$$

(ii) *for all $z \in \mathcal{I}$ with $\|M_{\mathcal{S}}(x) - M_{\mathcal{S}}(z)\|_\infty \leqslant \frac{m}{2n^{a+3}}$, $\mathrm{sp}(z) = \mathrm{sp}(x)$.*

Proof. Let y_i be the closest multiple of $n^{-(a+3)}$ to x_i with ties broken by preferring the smaller one. Then, y belongs to $\mathcal{I}$ and $\mathrm{sp}(y) = \mathrm{sp}(x)$ as $|x_i| \geqslant n^{-a}$ if $x_i \neq 0$. The inequality in (i) follows from the fact that

$$\left\| M_{\mathcal{S}}(x) - M_{\mathcal{S}}(y) \right\|_\infty \leqslant \sum_{i=1}^{n} |x_i - y_i| \leqslant \frac{m}{2n^{a+3}}.$$

For (ii), if $\|M_{\mathcal{S}}(z) - M_{\mathcal{S}}(x)\|_\infty \leqslant \frac{m}{2n^{a+3}}$, $\|M_{\mathcal{S}}(z) - M_{\mathcal{S}}(y)\|_\infty \leqslant \frac{m}{n^{a+3}}$ by triangle inequality. Since $\mathcal{S}$ separates $\mathcal{I}$ by more than $\frac{m}{n^{a+3}}$, $\mathrm{sp}(z)$ must be $\mathrm{sp}(y)$, which is $\mathrm{sp}(x)$. □

Lemmas 3.1 and 3.2 give Theorem 1.3 by the following algorithm.

Algorithm
// x is an unknown instance.
compute $M_{\mathcal{S}}(x)$;
for each $z \in \mathcal{I}$
  if $\|M_{\mathcal{S}}(x) - M_{\mathcal{S}}(z)\|_\infty \leqslant \frac{m}{2n^{a+3}}$,
    $y \leftarrow z$ and break;
return $\mathrm{sp}(y)$;

Now, we prove Lemma 3.1. Let $\mathcal{D}$ be the set of differences of instances in $\mathcal{I}$ with distinct supports, i.e., the set of vectors $x - y$ for all $x, y \in \mathcal{I}$ with $\mathrm{sp}(x) \neq \mathrm{sp}(y)$. Since $M_{\mathcal{S}}(\cdot)$ is linear for a fixed sequence $\mathcal{S}$, Lemma 3.1 is equivalent to saying that there exists a sequence $\mathcal{S}$ of ℓ queries such that $\|M_{\mathcal{S}}(x)\|_\infty > \frac{m}{n^{a+3}}$ for all $x \in \mathcal{D}$. A probabilistic method is used to prove the following lemma, which immediately implies Lemma 3.1. In the following, a uniform random query means a uniform random binary vector sampled from $\{0, 1\}^n$.

Lemma 3.3. *There exists a constant $c > 0$ such that*

$$\Pr\left[\textit{There is } x \textit{ in } \mathcal{D} \textit{ such that } \left\| M_{\mathcal{S}}(x) \right\|_\infty \leqslant \frac{m}{n^{a+3}}\right] = o(1)$$

for a sequence $\mathcal{S}$ of $\ell = \frac{cm \log n}{\log(m+1)}$ independent uniform random queries.

Notice that for all $x \in \mathcal{D}$, there is i such that $|x_i| \geqslant n^{-a}$ since x is the difference of a pair of instances with distinct supports. For a uniform random query q, since flipping the ith bit of q changes the answer $\chi_q(x)$ by at least n^{-a}, the probability of $|\chi_q(x)| < \frac{1}{2n^a}$ is at most 1/2. Thus, the probability of $\|M_{\mathcal{S}}(x)\|_\infty \leqslant \frac{m}{n^{a+3}} < \frac{1}{2n^a}$ is at most $2^{-\ell}$. On the other hand, the size $|\mathcal{D}|$ of $\mathcal{D}$ is at most $\sum_{i=1}^{2m} \binom{n}{i} (4n^{a+b+3})^i = e^{\mathcal{O}(m \log n)}$. If m is bounded from above, then we may take large enough c so that $2^{-\ell}|\mathcal{D}| = o(1)$. We may now assume that m is sufficiently large. It turns out that the bound $2^{-\ell}$ is not enough to use the union bound as the size of $\mathcal{D}$ is larger than 2^ℓ unless m is bounded from above. However, if there are many, say t, i's such that $|x_i| \geqslant \frac{m}{n^{a+3}}$, then the probability of $|\chi_q(x)| \leqslant \frac{m}{n^{a+3}}$ is $\mathcal{O}(\frac{1}{\sqrt{t}})$ by the generalized Littlewood–Offord theorem presented in Section 2. Observing this, we have the following lemma. Hereafter, "$P(m)$ holds for sufficiently large m" means that there is a constant $m_0 > 0$ such that $P(m)$ holds for all $m \geqslant m_0$.

Lemma 3.4. *For sufficiently large m, the following holds: Suppose that $x \in \mathcal{D}$ has at least $\frac{m}{\log(m+1)}$ coordinates larger than or equal to $\frac{m}{n^{a+3}}$ in absolute value and q is a uniform random query. Then,*

$$\Pr\left[\left|\chi_q(x)\right| \leqslant \frac{m}{n^{a+3}}\right] \leqslant m^{-0.4}.$$

Proof. Let X_i's be random variables such that $X_i = x_i$ if $q_i = 1$ and $X_i = 0$ otherwise. Then, the lemma immediately follows by the generalized Littlewood–Offord theorem as $\chi_q(x) = \sum_{i=1}^{n} X_i$, X_i's are independent, and, for at least $\frac{m}{\log(m+1)}$ i's, $|X_i| \geqslant \frac{m}{n^{a+3}}$ with probability $\frac{1}{2}$ while it is 0 with probability $\frac{1}{2}$. □

Let $\mathcal{D}_1$ be the set of all $x \in \mathcal{D}$ with less than $\frac{m}{\log(m+1)}$ coordinates larger than or equal to $\frac{m}{n^{a+3}}$ in absolute value and $\mathcal{D}_2 = \mathcal{D} \setminus \mathcal{D}_1$. Lemma 3.4 implies

$$\Pr\left[\text{There is } x \text{ in } \mathcal{D}_2 \text{ such that } \left\| M_{\mathcal{S}}(x) \right\|_\infty \leqslant \frac{m}{n^{a+3}}\right] \leqslant |\mathcal{D}| m^{-0.4\ell} = o(1) \tag{1}$$

for a large enough constant c in $\ell = \frac{cm \log n}{\log(m+1)}$. For $x \in \mathcal{D}_1$, let $\tilde{x}$ be the vector obtained by replacing the coordinates in x less than $\frac{m}{n^{a+3}}$ in absolute value with 0. Let $\tilde{\mathcal{D}}_1$ be the set of all such $\tilde{x}$. Notice that $\tilde{x} \in \tilde{\mathcal{D}}_1$ has at least one coordinate whose absolute value is at least n^{-a}. The same argument mentioned above gives $\Pr[\|M_{\mathcal{S}}(\tilde{x})\|_\infty < \frac{1}{2n^a}] \leqslant 2^{-\ell}$ and the union bound yields

$$\Pr\left[\text{There is } \tilde{x} \text{ in } \tilde{\mathcal{D}_1} \text{ such that } \left\|M_{\mathcal{S}}(\tilde{x})\right\|_\infty < \frac{1}{2n^a}\right] \leqslant |\tilde{\mathcal{D}_1}|2^{-\ell} \leqslant 2^{-\ell} \sum_{i=1}^{\lfloor \frac{m}{\log(m+1)} \rfloor} \binom{n}{i} \left(4n^{a+b+3}\right)^i = o(1)$$

for a large enough constant c in ℓ. Since

$$\left\|M_{\mathcal{S}}(x)\right\|_\infty \geqslant \left\|M_{\mathcal{S}}(\tilde{x})\right\|_\infty - \left\|M_{\mathcal{S}}(x-\tilde{x})\right\|_\infty \geqslant \left\|M_{\mathcal{S}}(\tilde{x})\right\|_\infty - \frac{2m^2}{n^{a+3}}$$

and $\frac{m}{n^{a+3}} + \frac{2m^2}{n^{a+3}} < \frac{1}{2n^a}$, we have

$$\Pr\left[\text{There is } x \text{ in } \mathcal{D}_1 \text{ such that } \left\|M_{\mathcal{S}}(x)\right\|_\infty \leqslant \frac{m}{n^{a+3}}\right] \leqslant \Pr\left[\text{There is } \tilde{x} \text{ in } \tilde{\mathcal{D}_1} \text{ such that } \left\|M_{\mathcal{S}}(\tilde{x})\right\|_\infty \leqslant \frac{m}{n^{a+3}} + \frac{2m^2}{n^{a+3}}\right] = o(1). \tag{2}$$

Lemma 3.3 follows by (1) and (2).

4. Finding graphs

In this section, we prove Theorem 1.1. Unlike the coin weighing problem, the number of all possible instances is finite as graphs are unweighted. Thus, a desired algorithm is immediately obtained by a sequence of cross-additive queries separating every pair of all possible graphs. To be more precise, let $\mathcal{G}$ be the set of all graphs on the vertex set $[n] = \{1, \ldots, n\}$ having at most m edges. For a graph $G \in \mathcal{G}$ and a sequence $\mathcal{S}$ of ℓ cross-additive queries, $M_{\mathcal{S}}(G)$ is the ℓ-dimensional vector consisting of the answers of the ℓ queries for G. The following implies Theorem 1.1.

Lemma 4.1. *There exists a sequence $\mathcal{S}$ of ℓ cross-additive queries such that $\ell = \mathcal{O}(\frac{m \log \frac{n^2}{m}}{\log(m+1)})$ and*

$$\left\|M_{\mathcal{S}}(G) - M_{\mathcal{S}}(H)\right\|_\infty > 0$$

for all distinct $G, H \in \mathcal{G}$.

We now prove Lemma 4.1. For two graphs $G, H \in \mathcal{G}$, define the difference graph $G - H$ to be the graph of which vertex set is $[n]$ and edge set is $(E(G) \setminus E(H)) \cup (E(H) \setminus E(G))$. Edges in $E(G) \setminus E(H)$ have weight 1 and edges in $E(H) \setminus E(G)$ have weight -1. Let $\mathcal{D}$ be the set of difference graphs $G - H$ for distinct $G, H \in \mathcal{G}$. As the answer of a query for a difference graph $G - H \in \mathcal{D}$ is the answer for G minus the answer for H,

$$M_{\mathcal{S}}(G - H) = M_{\mathcal{S}}(G) - M_{\mathcal{S}}(H)$$

for any sequence $\mathcal{S}$ of queries. Thus, Lemma 4.1 is equivalent to saying that there exists a sequence $\mathcal{S}$ of $\mathcal{O}(\frac{m \log(n^2/m)}{\log(m+1)})$ queries such that $\|M_{\mathcal{S}}(G)\|_\infty > 0$ for all $G \in \mathcal{D}$. A *random query* (S, T) means a cross-additive query for a pair of random sets of vertices S, T such that each vertex independently belongs to S, T, and none of S and T, each with probability $\frac{1}{3}$ so that $S \cap T = \emptyset$. We prove the following to obtain Lemma 4.1.

Lemma 4.2. *There exists a constant $c > 0$ such that*

$$\Pr\left[\text{There is } G \text{ in } \mathcal{D} \text{ such that } \left\|M_{\mathcal{S}}(G)\right\|_\infty = 0\right] = o(1)$$

for a sequence $\mathcal{S}$ of $\ell = \frac{cm \log(n^2/m)}{\log(m+1)}$ independent random queries.

Let $\mu_{S,T}(G)$ be the answer of a cross-additive query (S, T) for a graph G. For $G \in \mathcal{D}$, since G is the difference graph of two distinct graphs, G has at least one edge. By this fact, we have the following.

Lemma 4.3. *For $G \in \mathcal{D}$ and a random query (S, T),*

$$\Pr\left[\mu_{S,T}(G) = 0\right] \leqslant \frac{8}{9}.$$

Proof. Take an edge $e = \{x, y\}$ in G. We consider (S, T) conditioned on $\tilde{S} = S \cap ([n] \setminus \{x, y\})$ and $\tilde{T} = T \cap ([n] \setminus \{x, y\})$. In other words, all vertices except x and y are determined to be in S, T, or none of them. We show that $\mu_{S,T}(G) \neq 0$ when (S, T) is one of $(\tilde{S}, \tilde{T})$, $(\tilde{S} \cup \{x\}, \tilde{T})$, $(\tilde{S}, \tilde{T} \cup \{y\})$, and $(\tilde{S} \cup \{x\}, \tilde{T} \cup \{y\})$, each of which occurs with probability 1/9.

The sum of weights of edges crossing between $\tilde{S} \cup \{x\}$ and $\tilde{T}$ is that of edges crossing between $\tilde{S}$ and $\tilde{T}$ plus that of edges crossing between $\{x\}$ and $\tilde{T}$. Similarly, the sum of weights of edges crossing between $\tilde{S}$ and $\tilde{T} \cup \{y\}$ is that of edges crossing between $\tilde{S}$ and $\tilde{T}$ plus that of edges crossing between $\tilde{S}$ and $\{y\}$. Thus, letting $w_G(e)$ be the weight of e,

$$\mu_{\tilde{S}\cup\{x\},\tilde{T}\cup\{y\}}(G) = w_G(e) + \mu_{\tilde{S}\cup\{x\},\tilde{T}}(G) + \mu_{\tilde{S},\tilde{T}\cup\{y\}}(G) - \mu_{\tilde{S},\tilde{T}}(G).$$

If the last three terms are all 0, then $\mu_{\tilde{S}\cup\{x\},\tilde{T}\cup\{y\}}(G) = w_G(e) \neq 0$. $\square$

Thus, the probability of $\|M_{\mathcal{S}}(G)\|_\infty = 0$ is at most $(\frac{8}{9})^\ell$. As the size of $\mathcal{D}$ is at most $\sum_{i=1}^{2m} \binom{\binom{n}{2}}{i} 2^i = e^{\mathcal{O}(m \log(n^2/m))}$, we may take large enough c in $\ell = \frac{cm\log(n^2/m)}{\log(m+1)}$ so that $(8/9)^\ell |\mathcal{D}| = o(1)$ if m is bounded above. It may now be assumed that m is sufficiently large. The bound $(8/9)^\ell$ is not enough to use the union bound as the size of $\mathcal{D}$ is larger than $(9/8)^\ell$ unless m is bounded above. Notice that, for a random query (S, T), $\mu_{S,T}(G)$ is a sum of weights of edges in G chosen randomly. If the edges were chosen independently, one might hope to prove Lemma 4.2 by dividing $\mathcal{D}$ into two classes according to the number of edges as in the coin weighing problem. However, there is a dependency among edges in general and the graphs with many edges are further divided into subclasses to handle the dependency.

Precisely, we divide $\mathcal{D}$ into four classes $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$, and $\mathcal{D}_4$. The class $\mathcal{D}_1$ consists of graphs in $\mathcal{D}$ that have only few edges, namely, less than $\frac{m}{(\log(m+1))^2}$ edges. The class $\mathcal{D}_2$ consists of graphs in $\mathcal{D}$ that have many vertices of high degree. To be precise, let $\delta = 0.1$ and define $U(G)$ for a graph $G \in \mathcal{D}$ to be the set of vertices of degree at least m^δ in G. The class $\mathcal{D}_2$ is the set of graphs G in $\mathcal{D}$ with the size of $U(G)$ being m^δ or more. Define $W(G) = [n] \setminus U(G)$. The class $\mathcal{D}_3$ is the set of graphs G in $\mathcal{D} \setminus (\mathcal{D}_1 \cup \mathcal{D}_2)$ such that there are at least $\frac{m}{2(\log(m+1))^2}$ edges crossing between $U(G)$ and $W(G)$. The class $\mathcal{D}_4$ is $\mathcal{D} \setminus (\mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3)$. It turns out that, for $G \in \mathcal{D}_2$ or $\mathcal{D}_4$ and a random query (S, T), $\Pr[\mu_{S,T}(G) = 0]$ is polynomially small in m. On the other hand, it is not true for some graphs in $\mathcal{D}_3$ (e.g., the graphs G such that the size of $U(G)$ is a constant independent of n and there is no edge with both ends in $W(G)$). Instead of focusing on a single query, we consider queries collectively for graphs in $\mathcal{D}_3$. Except for a negligible probability, there is a constant portion of the ℓ random queries for each graph in $\mathcal{D}_3$ that give polynomially small bounds in m as shown below.

We consider four cases $G \in \mathcal{D}_i$, $i = 1, \ldots, 4$, and show that

$$\Pr\bigl[\text{There is } G \text{ in } \mathcal{D}_i \text{ such that } \|M_{\mathcal{S}}(G)\|_\infty = 0\bigr] = o(1)$$

for a large enough constant c in ℓ.

Case 1. $G \in \mathcal{D}_1$.

As the number of graphs in $\mathcal{D}_1$ is at most $\sum_{i=1}^{\lfloor \frac{m}{(\log(m+1))^2} \rfloor} \binom{\binom{n}{2}}{i} 2^i = e^{\mathcal{O}(\frac{m\log(n^2/m)}{\log(m+1)})}$, the desired inequality immediately follows by Lemma 4.3.

Case 2. $G \in \mathcal{D}_2$.

Clearly, there are at most $\sum_{i=1}^{2m} \binom{\binom{n}{2}}{i} 2^i = e^{\mathcal{O}(m \log(n^2/m))}$ graphs in $\mathcal{D}_2$. Thus, the desired inequality is obtained by the following lemma. The proof is based on two applications of the generalized Littlewood–Offord theorem.

Lemma 4.4. *For sufficiently large m, the following holds: Suppose that G is a graph in $\mathcal{D}_2$ and (S, T) is a random query. Then,*

$$\Pr\bigl[\mu_{S,T}(G) = 0\bigr] \leqslant m^{-0.01\delta}.$$

Proof. As $|U(G)| \geqslant m^\delta$, we may choose a subset Q of $U(G)$ with $|Q| = \lfloor m^{0.1\delta} \rfloor$. Denote by H the spanning subgraph of G consisting of the edges in G except those with both ends in Q. (A spanning subgraph is a subgraph containing all the vertices of the original graph.) The lemma follows by showing that

$$\Pr\bigl[\bigl|\mu_{S,T}(H)\bigr| \leqslant m^{0.2\delta}\bigr] \leqslant m^{-0.01\delta}$$

for a random query (S, T). This is because there are at most $\binom{|Q|}{2} \leqslant m^{0.2\delta}$ edges with both ends in Q so that $\mu_{S,T}(G) = 0$ implies $|\mu_{S,T}(H)| \leqslant m^{0.2\delta}$.

Given a pair of disjoint sets (A, B) of vertices in $\bar{Q} := [n] \setminus Q$, we say a vertex v in Q is *bad* for (A, B) if the sum of weights of edges crossing between A and v is less than $m^{0.2\delta}$ in absolute value, that is, $|\mu_{A,v}(H)| < m^{0.2\delta}$. (We simply write

$\mu_{A,v}(H)$ for $\mu_{A,\{v\}}(H)$.) A pair (A, B) is called *bad* if there is a bad vertex in Q for (A, B). It is *good* otherwise. Letting $(\tilde{S}, \tilde{T}) = (S \cap \bar{Q}, T \cap \bar{Q})$, we have

$$\Pr\big[\big|\mu_{S,T}(H)\big| \leqslant m^{0.2\delta}\big] = \sum_{(\tilde{S},\tilde{T})} \Pr\big[(\tilde{S}, \tilde{T})\big] \Pr\big[\big|\mu_{S,T}(H)\big| \leqslant m^{0.2\delta} \big| (\tilde{S}, \tilde{T})\big].$$

The right hand side may be upper bounded by

$$\Pr\big[(\tilde{S}, \tilde{T}) \text{ is bad}\big] + \sum_{(\tilde{S},\tilde{T}) \text{ good}} \Pr\big[(\tilde{S}, \tilde{T})\big] \Pr\big[\big|\mu_{S,T}(H)\big| \leqslant m^{0.2\delta} \big| (\tilde{S}, \tilde{T})\big].$$

Thus, it is enough to show that each of the above two terms is polynomially small in m, say $m^{-0.04\delta}$.

For a vertex $v \in Q$, $\mu_{\tilde{S},v}(H)$ is a sum of weights of edges $\{v, u\}$ with $u \in \bar{Q}$ chosen independently and with probability $\frac{1}{3}$. As v is adjacent to at least $m^\delta - |Q| \geqslant \frac{m^\delta}{2}$ vertices in $\bar{Q}$, the generalized Littlewood–Offord theorem gives

$$\Pr\big[v \text{ is bad for } (\tilde{S}, \tilde{T})\big] = \Pr\big[\big|\mu_{\tilde{S},v}(H)\big| < m^{0.2\delta}\big] \leqslant m^{-0.2\delta}.$$

Thus,

$$\Pr\big[(\tilde{S}, \tilde{T}) \text{ is bad}\big] \leqslant |Q| \cdot m^{-0.2\delta} \leqslant m^{-0.1\delta}.$$

Now, we show that, given the event that $(\tilde{S}, \tilde{T})$ is good, the probability of $|\mu_{S,T}(H)| \leqslant m^{0.2\delta}$ is polynomially small in m. To this end, we consider the condition that $(\tilde{S}, \tilde{T})$ is fixed to be a good pair. Since there is no edge of H with both ends in Q, $\mu_{S,T}(H)$ is decomposed into

$$\mu_{S,T}(H) = \mu_{S\cap Q,\tilde{T}}(H) + \mu_{\tilde{S},T\cap Q}(H) + \mu_{\tilde{S},\tilde{T}}(H).$$

For a vertex v in Q, let Y_v be a random variable such that $Y_v = \mu_{v,\tilde{T}}(H)$ if $v \in S$, $Y_v = \mu_{\tilde{S},v}(H)$ if $v \in T$, and $Y_v = 0$ otherwise. Then, $\mu_{S\cap Q,\tilde{T}}(H) + \mu_{\tilde{S},T\cap Q}(H) = \sum_{v\in Q} Y_v$ and

$$\mu_{S,T}(H) = \sum_{v\in Q} Y_v + \mu_{\tilde{S},\tilde{T}}(H).$$

Since there is no bad vertex of Q for $(\tilde{S}, \tilde{T})$, $|Y_v| \geqslant m^{0.2\delta}$ if $v \in T$, which occurs with probability $\frac{1}{3}$. On the other hand, $Y_v = 0$ if $v \notin S \cup T$, which also occurs with probability $\frac{1}{3}$. Since Y_v's are independent and the size of Q is $\lfloor m^{0.1\delta} \rfloor$, the probability that $|\sum_{v\in Q} Y_v + \mu_{\tilde{S},\tilde{T}}(H)| \leqslant m^{0.2\delta}$ is at most $m^{-0.04\delta}$ by the generalized Littlewood–Offord theorem. That is, for any good pair $(\tilde{S}, \tilde{T})$,

$$\Pr\big[\big|\mu_{S,T}(H)\big| \leqslant m^{0.2\delta} \big| (\tilde{S}, \tilde{T})\big] \leqslant m^{-0.04\delta}. \qquad \square$$

Case 3. $G \in \mathcal{D}_3$.

As mentioned above, a bound better than a constant may not be obtained for a random query for some graphs in $\mathcal{D}_3$. Taking this into account, we use a conditioning argument for the random queries in a collective sense to prove the desired inequality. We start with the following lemma.

Lemma 4.5. *For sufficiently large m, the following holds: Suppose that G is a graph in $\mathcal{D}_3$. Then, there is a pair of disjoint sets of vertices (A, B) such that*

(i) $0 < |A| \leqslant 2m^{2\delta}$ *and* $|B| = \lfloor m^\delta \rfloor$,
(ii) *for each* $v \in B$, *there is* $u \in A$ *that is adjacent to* v, *and*
(iii) *for all* $u \in B$ *and* $v \in [n] \setminus A$, u *and* v *are not adjacent.*

Proof. We prove the lemma by constructing the desired pair (A, B). As $G \in \mathcal{D}_3$, the number of edges crossing between $U(G)$ and $W(G)$ is at least $\frac{m}{2(\log(m+1))^2}$. Since each vertex in $W(G)$ is of degree less than m^δ, there are at least $(\frac{m}{2(\log(m+1))^2})/m^\delta \geqslant m^{8\delta}$ vertices in $W(G)$ that are adjacent to at least one vertex in $U(G)$. Thus, we may iteratively choose $\lfloor m^\delta \rfloor$ such vertices that are not adjacent to each other. This is possible as vertices in $W(G)$ are of degree less than m^δ and $m^{8\delta}/(1 + m^\delta) \geqslant m^\delta$. The set B consists of these vertices.

The set A consists of vertices in $U(G)$ and vertices in $W(G) \setminus B$ that are adjacent to a vertex in B. Since $G \notin \mathcal{D}_2$ and there are at most $|B|m^\delta$ vertices in $W(G) \setminus B$ described above, $|A| \leqslant |U(G)| + |B|m^\delta \leqslant 2m^{2\delta}$. Hence the properties (i)–(iii) follow. $\square$

For each graph G in $\mathcal{D}_3$, we choose a pair of disjoint sets of vertices (A, B) satisfying the properties in Lemma 4.5 and denote by $H(G)$ the bipartite subgraph of G on $A \cup B$ consisting of all edges in G between them. Recall that $\mathcal{S}$ is the sequence of random queries (S, T). Once all vertices in $\bar{B} := V(G) \setminus B$ are determined to be in S, T or none of them for the random queries (S, T), $M_{\mathcal{S}}(G[\bar{B}])$ is determined, where $G[\bar{B}]$ is the induced subgraph of G on $\bar{B}$. Also, $M_{\mathcal{S}}(G) = M_{\mathcal{S}}(H(G)) + M_{\mathcal{S}}(G[\bar{B}])$.

Let $\tilde{\mathcal{S}}$ be the sequence of restricted random queries $(\tilde{S}, \tilde{T})$ on $\bar{B}$. Then we may write $M_{\tilde{\mathcal{S}}}(G[\bar{B}])$ for $M_{\mathcal{S}}(G[\bar{B}])$ as $M_{\mathcal{S}}(G[\bar{B}])$ depends only on $\tilde{\mathcal{S}}$. The quantity $\|M_{\mathcal{S}}(H(G)) + M_{\tilde{\mathcal{S}}}(G[\bar{B}])\|_\infty$ turns out to be zero with small enough probability for most of $\tilde{\mathcal{S}}$. For the set $\mathcal{H}$ of all possible bipartite graphs H, say on $A \cup B$, satisfying (i) and (ii) in Lemma 4.5, we divide the event $\{\exists G \in \mathcal{D}_3$ such that $\|M_{\mathcal{S}}(G)\|_\infty = 0\}$ into events $F_H := \{\exists G \in \mathcal{D}_3$ with $H(G) = H$ such that $\|M_{\mathcal{S}}(G)\|_\infty = 0\}$, $H \in \mathcal{H}$. Observe that

$$\Pr\left[\exists G \in \mathcal{D}_3 \text{ such that } \left\|M_{\mathcal{S}}(G)\right\|_\infty = 0\right] = \Pr\left[\bigcup_{H \in \mathcal{H}} F_H\right] \leqslant \sum_{H \in \mathcal{H}} \Pr[F_H].$$

To estimate $\Pr[F_H]$, we further consider $\Pr[F_H] = \sum_{\tilde{\mathcal{S}}} \Pr[\tilde{\mathcal{S}}] \Pr[F_H | \tilde{\mathcal{S}}]$, and will show that $\Pr[F_H | \tilde{\mathcal{S}}]$ is small enough for most of $\tilde{\mathcal{S}}$.

For a given graph $H \in \mathcal{H}$ on $A \cup B$ and the restricted random query $(\tilde{S}, \tilde{T})$ on $\bar{B}$, a vertex $v \in B$ is called *bad*, if the sum of weights of edges between v and $\tilde{S} \cap A$ is 0, i.e., $\mu_{v, \tilde{S} \cap A}(H) = 0$. The restricted random query $(\tilde{S}, \tilde{T})$ is *bad* if more than $\frac{3}{4}|B|$ vertices of B are bad for $(\tilde{S}, \tilde{T})$. The sequence $\tilde{\mathcal{S}} = (\tilde{S}_i, \tilde{T}_i)_{i=1}^{\ell}$ is called *bad* if the number of bad queries $(\tilde{S}_i, \tilde{T}_i)$ is more than $\frac{9\ell}{10}$. The sequence is *good* if it is not bad. Observe that

$$\Pr[F_H] \leqslant \Pr[\tilde{\mathcal{S}} \text{ bad}] + \Pr[F_H \text{ and } \tilde{\mathcal{S}} \text{ good}] = \Pr[\tilde{\mathcal{S}} \text{ bad}] + \sum_{\tilde{\mathcal{S}} \text{ good}} \Pr[\tilde{\mathcal{S}}] \Pr[F_H | \tilde{\mathcal{S}}]. \tag{3}$$

We first show that $\Pr[\tilde{\mathcal{S}}$ bad$]$ is small enough.

Lemma 4.6. *For a given $H \in \mathcal{H}$ on $A \cup B$, there is a constant $\beta < 1$ such that* $\Pr[\tilde{\mathcal{S}}$ *bad*$] \leqslant \beta^{\ell}$.

Proof. We first show that a vertex $v \in B$ is bad with probability at most $2/3$. For each vertex $v \in B$, take $u \in A$ that is adjacent to v in H. Assuming $\tilde{S}_i \cap A \setminus \{u\}$ is determined, there are three possibilities for u, $u \in \tilde{S}_i$, $u \in \tilde{T}_i$ or $u \notin \tilde{S}_i \cup \tilde{T}_i$. If $\mu_{v, \tilde{S}_i \cap A \setminus \{u\}}(H) \neq 0$, then v is not bad provided $u \notin \tilde{S}_i \cup \tilde{T}_i$, which occurs with probability $1/3$. If $\mu_{v, \tilde{S}_i \cap A \setminus \{u\}}(H) = 0$, then $u \in \tilde{S}_i$ implies that $|\mu_{v, \tilde{S}_i \cap A}(H)| = 1$. As $u \in \tilde{S}_i$ with probability $1/3$, $v \in B$ is bad with probability at most $2/3$.

Therefore, the expected number of bad vertices in B is at most $\frac{2}{3}|B|$ and hence the probability that more than $3|B|/4$ vertices in B is bad is at most $8/9$ by Markov's inequality. This implies that the expected number of bad queries $(\tilde{S}_i, \tilde{T}_i)$ in $\tilde{\mathcal{S}}$ is less than or equal to $8\ell/9$. Since $(\tilde{S}_i, \tilde{T}_i)$ are i.i.d., Chernoff's large deviation inequality yields $\Pr[\tilde{\mathcal{S}}$ bad$] \leqslant \beta^{\ell}$, for a constant $\beta < 1$. $\square$

For an upper bound of the second part of (3), observe that

$$\Pr[F_H | \tilde{\mathcal{S}}] \leqslant \sum_{G: H(G) = H} \Pr\left[\left\|M_{\mathcal{S}}(H) + M_{\tilde{\mathcal{S}}}\left(G[\bar{B}]\right)\right\|_\infty = 0 \middle| \tilde{\mathcal{S}}\right]. \tag{4}$$

We show that

Lemma 4.7. *For sufficiently large m, the following holds: Suppose that $\tilde{\mathcal{S}}$ is a good sequence of restricted random queries $(\tilde{S}_i, \tilde{T}_i)$ on $\bar{B}$. Then,*

$$\Pr\left[\left\|M_{\mathcal{S}}(H) + M_{\tilde{\mathcal{S}}}\left(G[\bar{B}]\right)\right\|_\infty = 0 \middle| \tilde{\mathcal{S}}\right] \leqslant m^{-\delta\ell/30}.$$

Proof. Without loss of generality, it may be assumed that $(\tilde{S}_i, \tilde{T}_i)$ is good for $1 \leqslant i \leqslant \ell/10$. For each i in the range, we may choose a set B_i of $\frac{|B|}{4}$ good vertices in B. (Of course, it actually means $\lfloor |B|/4 \rfloor$.) Under the condition that $(\tilde{S}_i, \tilde{T}_i)$ is given and all the vertices in $B \setminus B_i$ have been determined to be in S_i, T_i, or none of them for the fully extended query (S_i, T_i), the event that a vertex $v \in B_i$ belongs to T_i (or none of S_i and T_i, resp.), which occurs with probability $1/3$, changes the sum of weights of edges crossing between S_i and T_i by at least 1 (or does not change the sum, resp.) independently of the other vertices in B_i. Thus, the generalized Littlewood–Offord theorem yields that the sum of weights of edges crossing between S_i and T_i is a fixed number with probability at most $|B_i|^{-0.4}$. Since $|B_i| = \frac{|B|}{4} \geqslant m^{0.9\delta}$ and (S_i, T_i), $1 \leqslant i \leqslant \frac{\ell}{10}$, are independent,

$$\Pr\left[\left\|M_{\mathcal{S}}(H) + M_{\tilde{\mathcal{S}}}\left(G[\bar{B}]\right)\right\|_\infty = 0 \middle| \tilde{\mathcal{S}}\right] \leqslant m^{-\delta\ell/30}. \qquad \square$$

Since there are at most $\sum_{i=1}^{2m}\binom{\binom{n}{2}}{i}2^i = e^{\mathcal{O}(m\log(n^2/m))}$ graphs in $\mathcal{D}_3$, the above lemma and (4) imply that

$$\Pr[F_H|\tilde{S}] \leqslant \exp\left(-\frac{\delta\ell\log m}{30} + \mathcal{O}\big(m\log(n^2/m)\big)\right) \leqslant \exp\big(-m\log(n^2/m)\big),$$

for large enough c in $\ell = \frac{cm\log(n^2/m)}{\log(m+1)}$. With this inequality and Lemma 4.6, (3) gives

$$\Pr[F_H] \leqslant \beta^{\ell/2},$$

and we finally have

$$\Pr\big[\exists G \in \mathcal{D}_3 \text{ such that } \|M_{\mathcal{S}}(G)\|_\infty = 0\big] \leqslant \sum_{H\in\mathcal{H}} \Pr[F_H] \leqslant |\mathcal{H}|\beta^{\ell/2}.$$

As the number of edges in a graph in $\mathcal{H}$ is at most $2m^{3\delta}$ and the number of possible edge weights (including 0) is at most 3, we have

$$|\mathcal{H}| \leqslant 3^{2m^{3\delta}}\binom{n}{\lfloor m^\delta\rfloor}\sum_{i=1}^{\lfloor 2m^{2\delta}\rfloor}\binom{n}{i} \leqslant \beta^{-\ell/4}$$

and the desired inequality follows.

Case 4. $G \in \mathcal{D}_4$.

For graphs $G \in \mathcal{D}_4$, we need two lemmas for the desired bound. The following is based on the fact that there are many edges with both ends in $W(G)$ while each vertex in $W(G)$ has a bounded degree.

Lemma 4.8. *For sufficiently large m, the following holds: Suppose that G is a graph in $\mathcal{D}_4$. Then, G has an induced matching consisting of $\lfloor m^\delta\rfloor$ edges.*

Proof. As $G \notin \mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3$, there are at least $\frac{m}{(\log(m+1))^2}$ edges in G (as $G \notin \mathcal{D}_1$) and there are less than $\frac{m}{2(\log(m+1))^2}$ edges crossing between $U(G)$ and $W(G)$ (as $G \notin \mathcal{D}_3$). Since the size of $U(G)$ is less than m^δ (as $G \notin \mathcal{D}_2$), there are at most $\binom{|U(G)|}{2} \leqslant \frac{m^{2\delta}}{2}$ edges with both ends in $U(G)$. Thus, there are at least $\frac{m}{4(\log(m+1))^2}$ edges with both ends in $W(G)$ for sufficiently large m. Considering edges with both ends in $W(G)$ only, iteratively take an edge e and delete e and all edges that share a vertex with e. Since each vertex in $W(G)$ has a degree less than m^δ, we are able to take at least

$$\left(\frac{m}{4(\log(m+1))^2}\right)\Big/\big(2m^\delta+1\big) \geqslant m^\delta$$

edges. The first $\lfloor m^\delta\rfloor$ such edges constitute an induced matching as desired. □

Since there are at most $\sum_{i=1}^{2m}\binom{\binom{n}{2}}{i}2^i = e^{\mathcal{O}(m\log(n^2/m))}$ different graphs in $\mathcal{D}_4$, the following lemma yields the desired bound. To prove the lemma, the above lemma and the generalized Littlewood–Offord theorem are used.

Lemma 4.9. *For sufficiently large m, the following holds: Suppose that G is a graph in $\mathcal{D}_4$ and (S,T) is a random query. Then,*

$$\Pr\big[\mu_{S,T}(G)=0\big] \leqslant m^{-0.4\delta}.$$

Proof. According to Lemma 4.8, G has an induced matching consisting of $\lfloor m^\delta\rfloor$ edges. Let B be the set of the vertices in the induced matching. We consider a random query (S,T) conditioned on $\tilde{S} := S\cap\bar{B}$, $\tilde{T} := T\cap\bar{B}$, where $\bar{B} := [n]\setminus B$. Let $e_1,\ldots,e_{\lfloor m^\delta\rfloor}$ be the $\lfloor m^\delta\rfloor$ edges in the induced matching. For each edge $e_i = \{u_i,v_i\}$, let X_i be the random variable representing the contribution of u_i and v_i to the value $\mu_{S,T}(G)$ so that

$$\mu_{S,T}(G) = \sum_{i=1}^{\lfloor m^\delta\rfloor} X_i + \mu_{\tilde{S},\tilde{T}}(G).$$

This is possible since e_i's are the only edges of G with both ends in B and they are pairwise disjoint. It is now possible to use the same argument used in Lemma 4.3. For each i, X_i may be 0, $\mu_{u_i,\tilde{T}}(G)$, $\mu_{\tilde{S},v_i}(G)$ or $\mu_{u_i,\tilde{T}}(G)+\mu_{\tilde{S},v_i}(G)+w_G(e_i)$ each with probability 1/9, corresponding to the events $\{u_i,v_i \notin S\cup T\}$, $\{u_i\in S, v_i\notin S\cup T\}$, $\{u_i\notin S\cup T, v_i\in T\}$, and $\{u_i\in S, v_i\in T\}$, respectively. Since $|w_G(e_i)|=1$, at least one of the last three values is 1/3 or more in absolute value. In particular, $X_i=0$, $|X_i|\geqslant 1/3$ each with probability at least 1/9. By the generalized Littlewood–Offord theorem, the probability that $\sum_{i=1}^{\lfloor m^\delta\rfloor} X_i + \mu_{\tilde{S},\tilde{T}}(G)$ is a fixed number is at most $m^{-0.4\delta}$ and hence so is the probability that $\mu_{S,T}(G)=0$. □

5. Finding weighted graphs

In this section, we prove Theorem 1.2. As in the coin weighing problem, we consider a set of discretized graphs to extend the idea for unweighted graphs to weighted ones. Precisely, let $\mathcal{G}$ be the set of all possible graphs considered in Theorem 1.2, i.e., the set of graphs on $[n]$ having at most m edges whose weights are between n^{-a} and n^b in absolute value. Let $\mathcal{I}$ be the set of graphs in $\mathcal{G}$ with weights in multiples of $n^{-(a+6)}$. A sequence $\mathcal{S}$ of queries *separates* a set of graphs if it separates every pair of graphs in the set with different sets of edges. We prove the existence of a sequence of queries separating $\mathcal{I}$ by more than $\frac{m}{n^{a+6}}$. As in the previous section, $M_{\mathcal{S}}(G)$ denotes the vector whose coordinates are given by the answers of a sequence $\mathcal{S}$ of queries for a graph G.

Lemma 5.1. *There exists a sequence $\mathcal{S}$ of ℓ cross-additive queries such that $\ell = \mathcal{O}(\frac{m\log n}{\log m})$ and*

$$\left\| M_{\mathcal{S}}(G) - M_{\mathcal{S}}(H)\right\|_\infty > \frac{m}{n^{a+6}}$$

for all $G, H \in \mathcal{I}$ with $E(G) \neq E(H)$.

Given such a sequence $\mathcal{S}$, the following holds. We omit the proof as it is essentially the same as that of Lemma 3.2.

Lemma 5.2. *Suppose that G is a graph in $\mathcal{G}$.*

(i) *There exists $H \in \mathcal{I}$ such that $E(G) = E(H)$ and*

$$\left\| M_{\mathcal{S}}(G) - M_{\mathcal{S}}(H)\right\|_\infty \leqslant \frac{m}{2n^{a+6}}.$$

(ii) *For all $I \in \mathcal{I}$ with $\| M_{\mathcal{S}}(G) - M_{\mathcal{S}}(I)\|_\infty \leqslant \frac{m}{2n^{a+6}}$, $E(I) = E(G)$.*

Lemmas 5.1 and 5.2 imply Theorem 1.2 by the following algorithm: Given an unknown graph G in $\mathcal{G}$, compute $M_{\mathcal{S}}(G)$ and find $H \in \mathcal{I}$ such that $\| M_{\mathcal{S}}(G) - M_{\mathcal{S}}(H)\|_\infty \leqslant \frac{m}{2n^{a+6}}$. The set of edges in H is the one that we are looking for. In this algorithm, $\mathcal{O}(\frac{m\log n}{\log m})$ queries are required only in computing $M_{\mathcal{S}}(G)$ and they are non-adaptive.

In the rest of this section, we prove Lemma 5.1. For two graphs $G, H \in \mathcal{I}$, define the difference graph $G - H$ to be the weighted graph of which vertex set is $[n]$ and edge set consists of those edges e with $w_G(e) - w_H(e) \neq 0$. Here, we use the convention that $w_G(e) = 0$ if e is not an edge of a weighted graph G. The weight of an edge e in $G - H$ is $w_G(e) - w_H(e)$. Let $\mathcal{D}$ be the set of difference graphs $G - H$ for $G, H \in \mathcal{I}$ with $E(G) \neq E(H)$. Then, the following lemma implies Lemma 5.1. As in the previous section, a *random query* (S, T) means a cross-additive query for a pair of random sets of vertices S, T such that each vertex independently belongs to S, T, and none of S and T, each with probability $\frac{1}{3}$ so that $S \cap T = \emptyset$.

Lemma 5.3 *(Main lemma). There exists a constant $c > 0$ such that*

$$\Pr\left[\text{There is } G \text{ in } \mathcal{D} \text{ such that } \left\| M_{\mathcal{S}}(G)\right\|_\infty \leqslant \frac{m}{n^{a+6}}\right] = o(1)$$

for a sequence $\mathcal{S}$ of $\ell = \frac{cm\log n}{\log m}$ independent random queries.

Notice that there may be edges of very small weight, e.g., $\frac{1}{n^{a+6}}$, in absolute value. For that reason, it seems hard to prove Lemma 5.3 by dividing the graphs in $\mathcal{D}$ regardless of weights of edges as in the proof for unweighted graphs. We divide $\mathcal{D}$ into four classes $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$, and $\mathcal{D}_4$ based on weights of edges. For the division of cases, we need to classify edges into three types according to their weights. An edge is called *heavy* if its weight is at least $\frac{1}{8mn^a}$ in absolute value. An edge is called *light* if its weight is less than $\frac{m}{n^{a+6}}$ in absolute value. An edge that is neither heavy nor light is called *middleweight*. The class $\mathcal{D}_1$ consists of graphs in $\mathcal{D}$ that have only few heavy edges, namely, less than $\frac{m}{\log m}$ heavy edges. The class $\mathcal{D}_2$ consists of graphs in $\mathcal{D}$ that have many vertices incident with many heavy or middleweight edges. To be precise, let $\delta = 0.1$ and define $U(G)$ for a weighted graph G to be the set of the vertices that are incident with at least m^δ heavy or middleweight edges. The class $\mathcal{D}_2$ is the set of graphs G in $\mathcal{D}$ with the size of $U(G)$ being m^δ or more. Define $W(G) = [n] \setminus U(G)$. The class $\mathcal{D}_3$ is the set of graphs G in $\mathcal{D} \setminus (\mathcal{D}_1 \cup \mathcal{D}_2)$ such that there are at least $\frac{m}{2\log m}$ heavy edges crossing between $U(G)$ and $W(G)$. The class $\mathcal{D}_4$ is $\mathcal{D} \setminus (\mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3)$. We consider four cases $G \in \mathcal{D}_i$, $i = 1, \ldots, 4$, and show that

$$\Pr\left[\text{There is } G \text{ in } \mathcal{D}_i \text{ such that } \left\| M_{\mathcal{S}}(G)\right\|_\infty \leqslant \frac{m}{n^{a+6}}\right] = o(1)$$

provided $m \geqslant (\log n)^\alpha$ for a large constant α. The condition for the range of m is only used to derive the result for the second case, $G \in \mathcal{D}_2$ (see Lemma 5.4).

Case 1. $G \in \mathcal{D}_1$.

For $G \in \mathcal{D}_1$, let $\tilde{G}$ be the graph obtained by removing the middleweight and light edges from G. Let $\tilde{\mathcal{D}}_1$ denote the set of such graphs $\tilde{G}$. It turns out that if $\mathcal{S}$ separates $\tilde{\mathcal{D}}_1$ by at least $\frac{1}{4n^a}$, $\mathcal{S}$ separates $\mathcal{D}_1$ by more than $\frac{m}{n^{a+6}}$. We first derive an inequality for $\tilde{\mathcal{D}}_1$. Notice that for $G \in \mathcal{D}_1$, there is an edge in G of weight at least n^{-a} in absolute value as G is a difference graph of two graphs in $\mathcal{I}$ with different sets of edges. Thus, every $H \in \tilde{\mathcal{D}}_1$ has such an edge and $\Pr[|\mu_{S,T}(H)| < \frac{1}{4n^a}]$ for a random query (S,T) is at most 8/9 by the same argument used in Lemma 4.3. As the size of $\tilde{\mathcal{D}}_1$ is at most $\sum_{i=1}^{\lfloor \frac{m}{\log m} \rfloor} \binom{\binom{n}{2}}{i}(4n^{a+b+6})^i = e^{\mathcal{O}(\frac{m\log n}{\log m})}$, the union bound yields

$$\Pr\left[\text{There is } H \text{ in } \tilde{\mathcal{D}}_1 \text{ such that } \left\|M_{\mathcal{S}}(H)\right\|_\infty < \frac{1}{4n^a}\right] \leqslant |\tilde{\mathcal{D}}_1|\left(\frac{8}{9}\right)^{\ell} = o(1) \tag{5}$$

for a large enough constant c in $\ell = \frac{cm\log n}{\log m}$.

Since $G - \tilde{G}$ consists of at most $2m-1$ middleweight or light edges,

$$\left\|M_{\mathcal{S}}(G)\right\|_\infty \geqslant \left\|M_{\mathcal{S}}(\tilde{G})\right\|_\infty - \left\|M_{\mathcal{S}}(G-\tilde{G})\right\|_\infty \geqslant \left\|M_{\mathcal{S}}(\tilde{G})\right\|_\infty - \frac{2m-1}{8mn^a}.$$

The fact that $\frac{m}{n^{a+6}} + \frac{2m-1}{8mn^a} < \frac{1}{4n^a}$ and (5) imply

$$\begin{aligned}&\Pr\left[\text{There is } G \text{ in } \mathcal{D}_1 \text{ such that } \left\|M_{\mathcal{S}}(G)\right\|_\infty \leqslant \frac{m}{n^{a+6}}\right] \\ &\quad\leqslant \Pr\left[\text{There is } \tilde{G} \text{ in } \tilde{\mathcal{D}}_1 \text{ such that } \left\|M_{\mathcal{S}}(\tilde{G})\right\|_\infty \leqslant \frac{m}{n^{a+6}} + \frac{2m-1}{8mn^a}\right] = o(1).\end{aligned}$$

Case 2. $G \in \mathcal{D}_2$.

The proof for $\mathcal{D}_2$ is analogous to that of the second case for unweighted graphs. Let us first show that there is a set of vertices incident with many edges of weights in some range while the weights of edges in the set are properly bounded.

Lemma 5.4. *For some constants $\alpha > 0$ and $m_0 > 0$, the following holds provided $m \geqslant m_0$ and $m \geqslant (\log n)^\alpha$: Suppose that G is a graph in $\mathcal{D}_2$. Then, for $s = \frac{m}{n^{a+6}}$, there exist $Q \subseteq [n]$ and $i \geqslant 0$ such that*

(i) *the size of Q is $\lfloor m^{0.1\delta} \rfloor$,*
(ii) *each edge with both ends in Q has a weight less than $2^{i+1}s$ in absolute value, and*
(iii) *each vertex in Q is incident with at least $m^{0.8\delta}$ edges with weights in $[2^i s, 2^{i+1}s)$ in absolute value.*

Proof. An edge in G is called *type i* if the absolute value of its weight is in $[2^i s, 2^{i+1}s)$. For each vertex v of G, denote $d_i(v)$ to be the number of edges of type i containing v. Notice that a vertex $v \in U(G)$ is contained in at least m^δ edges with weights $s = \frac{m}{n^{a+6}}$ or more in absolute value and all weights of edges are at most $2n^b$ in absolute value. Let t be the minimum integer such that $2^t s > 2n^b$. Then, $\sum_{i=0}^{t-1} d_i(v) \geqslant m^\delta$ and $t = \mathcal{O}(\log n)$. Hence, we choose a large enough $\alpha > 0$ so that there is an integer i with $0 \leqslant i \leqslant t-1$ such that $d_i(v) \geqslant m^{0.8\delta}$ if $m \geqslant (\log n)^\alpha$. We decompose $U(G)$ into $U_0, \ldots, U_{t-1}$ so that $v \in U_i$ for the largest i with $d_i(v) \geqslant m^{0.8\delta}$. As the size of $U(G)$ is at least m^δ, there is i such that the size of U_i is at least $m^{0.95\delta}$. Notice that all vertices in U_i satisfy the property (iii).

To construct the desired vertex set Q, we iteratively take a vertex u in U_i and delete u and all vertices in U_i that are adjacent to u by an edge of type j for some $j > i$. Then, each time at most $(t-i-1)m^{0.8\delta} + 1 \leqslant m^{0.85\delta}$ vertices in U_i are to be deleted and hence we are able to take $\lfloor m^{0.1\delta} \rfloor$ vertices in U_i, which is the property (i). The property (ii) follows by the construction. □

Clearly, there are at most $\sum_{i=1}^{2m} \binom{\binom{n}{2}}{i}(4n^{a+b+6})^i = e^{\mathcal{O}(m\log n)}$ graphs in $\mathcal{D}_2$. The desired bound for $\mathcal{D}_2$ is obtained by the following lemma as $\ell = \frac{cm\log n}{\log m}$. The proof uses Lemma 5.4 and is essentially the same as that of Lemma 4.4.

Lemma 5.5. *For some constants $\alpha > 0$ and $m_0 > 0$, the following holds provided $m \geqslant m_0$ and $m \geqslant (\log n)^\alpha$: Suppose that G is a graph in $\mathcal{D}_2$ and (S,T) is a random query. Then,*

$$\Pr\left[\left|\mu_{S,T}(G)\right| \leqslant \frac{m}{n^{a+6}}\right] \leqslant m^{-0.01\delta}.$$

Proof. For $G \in \mathcal{D}_2$, we choose Q and i satisfying the properties in Lemma 5.4. Denote by H the spanning subgraph of G consisting of the edges in G except those with both ends in Q. By (i) and (ii) of Lemma 5.4, the sum of weights of edges with both ends in Q is at most $\binom{|Q|}{2}2^{i+1}s \leqslant 2^i sm^{0.2\delta}$ in absolute value. Thus, $|\mu_{S,T}(G)| \leqslant \frac{m}{n^{a+6}} = s$ implies

$$\left|\mu_{S,T}(H)\right| \leqslant \left|\mu_{S,T}(G)\right| + \left|\mu_{S,T}(G-H)\right| \leqslant s + 2^i sm^{0.2\delta} \leqslant 2^{i+1} sm^{0.2\delta}$$

as $i \geqslant 0$. To obtain the inequality, it is enough to show

$$\Pr\left[\left|\mu_{S,T}(H)\right| \leqslant 2^{i+1} sm^{0.2\delta}\right] \leqslant m^{-0.01\delta}.$$

For a pair of disjoint sets (A, B) of vertices in $\bar{Q} := [n] \setminus Q$, we say a vertex $v \in Q$ is *bad* for (A, B) if the sum of weights of edges in H crossing between A and v is less than $2^{i+1}sm^{0.2\delta}$ in absolute value, that is, $|\mu_{A,v}(H)| < 2^{i+1}sm^{0.2\delta}$. A pair (A, B) is called *bad* if there is a bad vertex in Q for (A, B). It is *good* otherwise. Letting $(\tilde{S}, \tilde{T}) = (S \cap \bar{Q}, T \cap \bar{Q})$, $\Pr[|\mu_{S,T}(H)| \leqslant 2^{i+1}sm^{0.2\delta}]$ is upper bounded by

$$\Pr\left[(\tilde{S}, \tilde{T}) \text{ is bad}\right] + \sum_{(\tilde{S}, \tilde{T}) \text{ good}} \Pr\left[(\tilde{S}, \tilde{T})\right] \Pr\left[\left|\mu_{S,T}(H)\right| \leqslant 2^{i+1}sm^{0.2\delta} \middle| (\tilde{S}, \tilde{T})\right].$$

Thus, it is enough to show that each of the above two terms is polynomially small in m, say $m^{-0.04\delta}$.

For a vertex $v \in Q$, $\mu_{\tilde{S},v}(H)$ is a sum of weights of edges $\{v, u\}$ with $u \in \bar{Q}$ chosen independently and with probability $\frac{1}{3}$. As there are at least $m^{0.8\delta} - |Q| \geqslant \frac{m^{0.8\delta}}{2}$ edges $\{v, u\}$ with $|w_H(uv)| \geqslant 2^i s$, the generalized Littlewood–Offord theorem gives

$$\Pr\left[v \text{ is bad for } (\tilde{S}, \tilde{T})\right] = \Pr\left[\left|\mu_{\tilde{S},v}(H)\right| < 2^{i+1}sm^{0.2\delta}\right] \leqslant m^{-0.15\delta}.$$

Thus, the union bound yields

$$\Pr\left[\text{There is a bad vertex in } Q \text{ for } (\tilde{S}, \tilde{T})\right] \leqslant |Q| \cdot m^{-0.15\delta} \leqslant m^{-0.05\delta}.$$

Now, we show that, given the condition that $(\tilde{S}, \tilde{T})$ is a fixed pair without a bad vertex in Q, the probability of $|\mu_{S,T}(H)| \leqslant 2^{i+1}sm^{0.2\delta}$ is polynomially small in m. Since there is no edge of H with both ends in Q, $\mu_{S,T}(H)$ is $\mu_{\tilde{S},\tilde{T}}(H)$ plus a sum of contributions Y_v of v over $v \in Q$ such that Y_v is $\mu_{v,\tilde{T}}(H)$, $\mu_{\tilde{S},v}(H)$, and 0 when v belongs to S, T, and none of them, respectively. The contribution Y_v changes at least $|\mu_{\tilde{S},v}(H)| \geqslant 2^{i+1}sm^{0.2\delta}$ according to $v \in T$ or $v \notin S \cup T$, each of which occurs with probability $\frac{1}{3}$. As Y_v's are independent for fixed $(\tilde{S}, \tilde{T})$ and the size of Q is $\lfloor m^{0.1\delta} \rfloor$, the probability of $|\mu_{S,T}(H)| \leqslant 2^{i+1}sm^{0.2\delta}$ is at most $m^{-0.04\delta}$ by the generalized Littlewood–Offord theorem. That is, for any good pair $(\tilde{S}, \tilde{T})$,

$$\Pr\left[\left|\mu_{S,T}(H)\right| \leqslant 2^{i+1}sm^{0.2\delta} \middle| (\tilde{S}, \tilde{T})\right] \leqslant m^{-0.04\delta}. \qquad \square$$

Case 3. $G \in \mathcal{D}_3$.

The proof is analogous to that of the third case for unweighted graphs. We start with the following lemma similar to Lemma 4.5.

Lemma 5.6. *For sufficiently large m, the following holds: Suppose that G is a graph in $\mathcal{D}_3$. Then, there is a pair of disjoint sets of vertices (A, B) such that*

(i) $0 < |A| \leqslant 2m^{2\delta}$ *and* $|B| = \lfloor m^\delta \rfloor$,
(ii) *for each $v \in B$, there is $u \in A$ that is adjacent to v by a heavy edge, and*
(iii) *for all $u \in B$ and $v \in [n] \setminus A$, u and v are joined only by a light edge.*

Proof. We construct the desired pair (A, B). As $G \in \mathcal{D}_3$, the number of heavy edges crossing between $U(G)$ and $W(G)$ is at least $\frac{m}{2\log m}$. Since each vertex in $W(G)$ is incident to less than m^δ heavy edges, there are $(\frac{m}{2\log m})/m^\delta \geqslant m^{8\delta}$ vertices in $W(G)$ that are adjacent to at least one vertex in $U(G)$ by a heavy edge in G. Thus, we may iteratively choose $\lfloor m^\delta \rfloor$ such vertices that are not adjacent to each other by heavy or middleweight edges, as vertices in $W(G)$ are incident to at most m^δ heavy or middleweight edges and $m^{8\delta}/(1 + m^\delta) \geqslant m^\delta$. The set B consists of these vertices.

The set A consists of vertices in $U(G)$ and vertices in $W(G) \setminus B$ that are adjacent to a vertex in B by a heavy or middleweight edge. Since $G \notin \mathcal{D}_2$ and there are at most $|B|m^\delta$ vertices in $W(G) \setminus B$ described above, $|A| \leqslant |U(G)| + |B|m^\delta \leqslant 2m^{2\delta}$. Hence the properties (i)–(iii) follow. $\square$

For each graph G in $\mathcal{D}_3$, we choose a pair of disjoint sets of vertices (A, B) satisfying the properties in Lemma 5.6 and $H(G)$ denotes the bipartite subgraph of G on $A \cup B$ consisting of all edges in G between them. Recall that $\mathcal{S}$ is the sequence of random queries (S, T). Once all vertices in $\bar{B} := V(G) \setminus B$ are determined to be in S, T or none of them for the random queries (S,T), $M_{\mathcal{S}}(G[\bar{B}])$ is determined, where $G[\bar{B}]$ is the induced subgraph of G on $\bar{B}$. It is expected that $\|M_{\mathcal{S}}(H(G)) + M_{\mathcal{S}}(G[\bar{B}])\|_\infty$ plays a major role in $\|M_{\mathcal{S}}(G)\|_\infty$. This is so since

$$\begin{aligned}\big\|M_{\mathcal{S}}(G)\big\|_\infty &\geqslant \big\|M_{\mathcal{S}}\big(H(G)\big) + M_{\mathcal{S}}\big(G[\bar{B}]\big)\big\|_\infty - \frac{m}{n^{a+6}}\left(\frac{|B|^2}{4} + |B|n\right) \\ &\geqslant \big\|M_{\mathcal{S}}\big(H(G)\big) + M_{\mathcal{S}}\big(G[\bar{B}]\big)\big\|_\infty - \frac{1}{10n^{a+2}}.\end{aligned} \tag{6}$$

Let $\tilde{\mathcal{S}}$ be the sequence of restricted random queries $(\tilde{S},\tilde{T})$ on $\bar{B}$. Then $M_{\mathcal{S}}(G[\bar{B}])$ is $M_{\tilde{\mathcal{S}}}(G[\bar{B}])$ as $M_{\mathcal{S}}(G[\bar{B}])$ depends only on $\tilde{\mathcal{S}}$. The quantity $\|M_{\mathcal{S}}(H(G)) + M_{\tilde{\mathcal{S}}}(G[\bar{B}])\|_\infty$ turns out to be large, namely at least $\frac{1}{8mn^a} > \frac{1}{10n^{a+2}} + \frac{m}{n^{a+6}}$, except for a small enough probability for most of $\tilde{\mathcal{S}}$. For the set $\mathcal{H}$ of all possible weighted bipartite graphs H, say on $A \cup B$, satisfying (i) and (ii) in Lemma 5.6, we divide the event $\{\exists G \in \mathcal{D}_3$ such that $\|M_{\mathcal{S}}(G)\|_\infty \leqslant \frac{m}{n^{a+6}}\}$ into events $F_H := \{\exists G \in \mathcal{D}_3$ with $H(G) = H$ such that $\|M_{\mathcal{S}}(G)\|_\infty \leqslant \frac{m}{n^{a+6}}\}$, $H \in \mathcal{H}$. Then,

$$\Pr\left[\exists G \in \mathcal{D}_3 \text{ such that } \big\|M_{\mathcal{S}}(G)\big\|_\infty \leqslant \frac{m}{n^{a+6}}\right] = \Pr\left[\bigcup_{H\in\mathcal{H}} F_H\right] \leqslant \sum_{H\in\mathcal{H}} \Pr[F_H].$$

To bound $\Pr[F_H]$, we further consider $\Pr[F_H] = \sum_{\tilde{\mathcal{S}}} \Pr[\tilde{\mathcal{S}}]\Pr[F_H|\tilde{\mathcal{S}}]$, and will show that $\Pr[F_H|\tilde{\mathcal{S}}]$ is small enough for most of $\tilde{\mathcal{S}}$.

For a given graph $H \in \mathcal{H}$ on $A \cup B$ and the restricted random query $(\tilde{S},\tilde{T})$ on $\bar{B}$, a vertex $v \in B$ is called *bad*, if the sum of weights of edges crossing between v and $\tilde{S} \cap A$ is negligible, i.e., $|\mu_{v,\tilde{S}\cap A}(H)| < \frac{1}{16mn^a}$. The restricted random query $(\tilde{S},\tilde{T})$ is *bad* if more than $\frac{3}{4}|B|$ vertices of B are bad for it. The sequence $\tilde{\mathcal{S}} = (\tilde{S}_i,\tilde{T}_i)_{i=1}^{\ell}$ is called *bad* if the number of bad queries $(\tilde{S}_i,\tilde{T}_i)$ is more than $\frac{9\ell}{10}$. The sequence is *good* if it is not bad. We have

$$\Pr[F_H] \leqslant \Pr[\tilde{\mathcal{S}} \text{ bad}] + \Pr[F_H \text{ and } \tilde{\mathcal{S}} \text{ good}] = \Pr[\tilde{\mathcal{S}} \text{ bad}] + \sum_{\tilde{\mathcal{S}} \text{ good}} \Pr[\tilde{\mathcal{S}}]\Pr[F_H|\tilde{\mathcal{S}}]. \tag{7}$$

Let us first show that $\Pr[\tilde{\mathcal{S}} \text{ bad}]$ is small enough.

Lemma 5.7. *For a given $H \in \mathcal{H}$ on $A \cup B$, there is a constant $\beta < 1$ such that* $\Pr[\tilde{\mathcal{S}}$ *bad*$] \leqslant \beta^\ell$.

Proof. For a vertex $v \in B$,

$$\Pr[v \text{ is bad}] = \Pr\left[\big|\mu_{v,\tilde{S}\cap A}(H)\big| < \frac{1}{16mn^a}\right] \leqslant \frac{2}{3}$$

as v is adjacent to a vertex $u \in A$ by a heavy edge and so $|\mu_{v,\tilde{S}\cap A}(H)|$ changes by at least $\frac{1}{8mn^a}$ according to $u \in \tilde{S}$ or not. Thus, the expected number of bad vertices in B is at most $\frac{2}{3}|B|$ and hence the probability that more than $3|B|/4$ vertices in B is bad is at most $8/9$ by Markov's inequality. This implies that the expected number of bad queries $(\tilde{S}_i,\tilde{T}_i)$ in $\tilde{\mathcal{S}}$ is less than or equal to $8\ell/9$. Since $(\tilde{S}_i,\tilde{T}_i)$ are i.i.d., Chernoff's large deviation inequality yields $\Pr[\tilde{\mathcal{S}} \text{ bad}] \leqslant \beta^\ell$ for a constant $\beta < 1$. □

To bound the second part of (7), notice that (6) and $\frac{1}{8mn^a} > \frac{1}{10n^{a+2}} + \frac{m}{n^{a+6}}$ imply

$$\Pr[F_H|\tilde{\mathcal{S}}] \leqslant \sum_{G:H(G)=H} \Pr\left[\big\|M_{\mathcal{S}}(H) + M_{\tilde{\mathcal{S}}}\big(G[\bar{B}]\big)\big\|_\infty \leqslant \frac{1}{8mn^a}\bigg|\tilde{\mathcal{S}}\right]. \tag{8}$$

We show

Lemma 5.8. *For sufficiently large m, the following holds: Suppose that $\tilde{\mathcal{S}}$ is a good sequence of restricted random queries $(\tilde{S}_i,\tilde{T}_i)$ on $\bar{B}$. Then,*

$$\Pr\left[\big\|M_{\mathcal{S}}(H) + M_{\tilde{\mathcal{S}}}\big(G[\bar{B}]\big)\big\|_\infty \leqslant \frac{1}{8mn^a}\bigg|\tilde{\mathcal{S}}\right] \leqslant m^{-\delta\ell/30}.$$

Proof. Suppose that $(\tilde{S}_i,\tilde{T}_i)$ is good and consider the fully extended random query (S_i,T_i). Then, there are at least $\lfloor\frac{|B|}{4}\rfloor \geqslant m^{0.9\delta}$ good vertices in B and for a good vertex $v \in B$, the answer of the query (S_i,T_i) changes by at least $\frac{1}{16mn^a}$ depending on whether $v \in T$ or $v \notin S \cup T$, each of which occurs with probability $\frac{1}{3}$, independently of the other vertices in B. Thus, the answer of (S_i, T_i) is at most $\frac{1}{8mn^a}$ in absolute value with probability $m^{-0.4\delta}$ or less by the generalized Littlewood–Offord theorem. Since at least $\ell/10$ $(\tilde{S}_i, \tilde{T}_i)$'s are good and (S_i, T_i)'s are independent, the lemma follows. □

Since there are $e^{\mathcal{O}(m\log n)}$ graphs in $\mathcal{D}_3$, the above lemma and (8) imply that

$$\Pr[F_H|\tilde{\mathcal{S}}] \leqslant \exp\left(-\frac{\delta\ell\log m}{30} + \mathcal{O}(m\log n)\right) \leqslant \exp(-m\log n),$$

for large enough c in $\ell = \frac{cm\log n}{\log m}$. With this inequality and Lemma 5.7, (7) gives $\Pr[F_H] \leqslant \beta^{\ell/2}$ for sufficiently large n and we finally have

$$\Pr\left[\exists G \in \mathcal{D}_3 \text{ such that } \left\|M_{\mathcal{S}}(G)\right\|_\infty \leqslant \frac{m}{n^{a+6}}\right] \leqslant \sum_{H\in\mathcal{H}} \Pr[F_H] \leqslant |\mathcal{H}|\beta^{\ell/2}.$$

Since, for the graphs in $\mathcal{H}$, the number of edges is at most $2m^{3\delta}$ and the number of possible edge weights (including 0) is at most $4n^{b+a+6}$, we have

$$|\mathcal{H}| \leqslant \left(4n^{b+a+6}\right)^{2m^{3\delta}} \binom{n}{m^\delta} \sum_{i=1}^{2m^{2\delta}} \binom{n}{i} \leqslant \beta^{-\ell/4}$$

and the desired inequality follows.

Case 4. $G \in \mathcal{D}_4$.

The proof is analogous to that of the fourth case for unweighted graphs. Let us first show that for $G \in \mathcal{D}_4$, there is a matching of a number of heavy edges if we do not consider the light edges in G.

Lemma 5.9. *For sufficiently large m, the following holds: Suppose that G is a graph in $\mathcal{D}_4$. Then, G has $\lfloor m^\delta \rfloor$ pairwise disjoint heavy edges (regarding edges as sets of two vertices) such that any other edge joining two vertices in the heavy edges is light.*

Proof. Notice that, as $G \notin \mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3$, there are at least $\frac{m}{4\log m}$ heavy edges with both ends in $W(G)$ and each vertex in $W(G)$ is incident with less than m^δ heavy or middleweight edges. Thus, we are able to iteratively choose a heavy edge e and delete e and all the heavy and middleweight edges sharing a vertex with e at least

$$\left(\frac{m}{4\log m}\right)/\left(2m^\delta + 1\right) \geqslant m^\delta$$

times. The first $\lfloor m^\delta \rfloor$ edges chosen are the desired edges. □

Since there are $e^{\mathcal{O}(m\log n)}$ graphs in $\mathcal{D}_4$, the following lemma yields the desired bound.

Lemma 5.10. *For sufficiently large m, the following holds: Suppose that G is a graph in $\mathcal{D}_4$ and (S, T) is a random query. Then,*

$$\Pr\left[\left|\mu_{S,T}(G)\right| \leqslant \frac{m}{n^{a+6}}\right] \leqslant m^{-0.4\delta}.$$

Proof. According to Lemma 5.9, we may take $\lfloor m^\delta \rfloor$ pairwise disjoint heavy edges $e_1, \dots, e_{\lfloor m^\delta \rfloor}$ of G such that any other edge joining two vertices contained in the heavy edges is light. Let B be the set of the vertices contained in the heavy edges. Define H to be the spanning subgraph of G consisting of the edges in G except for the light edges with both ends in B. Since the graph $G - H$ has only the light edges with both ends in B and $|B| = 2\lfloor m^\delta \rfloor$,

$$\left|\mu_{S,T}(G-H)\right| \leqslant \binom{2m^\delta}{2} \frac{m}{n^{a+6}}$$

and so $|\mu_{S,T}(G)| \leqslant \frac{m}{n^{a+6}}$ implies

$$\left|\mu_{S,T}(H)\right| \leqslant \left|\mu_{S,T}(G)\right| + \left|\mu_{S,T}(G-H)\right| \leqslant \left(\binom{2m^\delta}{2} + 1\right)\frac{m}{n^{a+6}} < \frac{1}{8mn^a}.$$

To obtain the inequality, it is enough to show that

$$\Pr\left[\left|\mu_{S,T}(H)\right| < \frac{1}{8mn^a}\right] \leqslant m^{-0.4\delta}.$$

We consider a random query (S,T) conditioned on $\tilde{S} := S \cap \bar{B}$, $\tilde{T} := T \cap \bar{B}$, where $\bar{B} := [n] \setminus B$. That is, all vertices in $\bar{B}$ have been determined to be in S, T, or none of them. For each heavy edge $e_i = \{u_i, v_i\}$ in B, let X_i be the random variable representing the contribution of u_i and v_i to the value $\mu_{S,T}(H)$ so that

$$\mu_{S,T}(H) = \sum_{i=1}^{\lfloor m^\delta \rfloor} X_i + \mu_{\tilde{S},\tilde{T}}(H).$$

This is possible since e_i's are the only edges of H with both ends in B and they are pairwise disjoint. For each i, X_i may be 0, $\mu_{u_i,\tilde{T}}(H)$, $\mu_{\tilde{S},v_i}(H)$ or $\mu_{u_i,\tilde{T}}(H) + \mu_{\tilde{S},v_i}(H) + w_G(e_i)$ each with probability 1/9, corresponding to the events $\{u_i, v_i \notin S \cup T\}$, $\{u_i \in S, v_i \notin S \cup T\}$, $\{u_i \notin S \cup T, v_i \in T\}$ and $\{u_i \in S, v_i \in T\}$, respectively. Since e_i is heavy, i.e., $|w_G(e_i)| \geqslant \frac{1}{8mn^a}$, at least one of the last three values is $\frac{1}{24mn^a}$ or more in absolute value. In particular, $X_i = 0$, $|X_i| \geqslant \frac{1}{24mn^a}$ each with probability at least 1/9. By the generalized Littlewood–Offord theorem, the probability that $\sum_{i=1}^{\lfloor m^\delta \rfloor} X_i + \mu_{\tilde{S},\tilde{T}}(H)$ is in a fixed interval of length $\frac{1}{4mn^a}$ is at most $m^{-0.4\delta}$ and hence so is the probability that $|\mu_{S,T}(H)| < \frac{1}{8mn^a}$. □

6. Finding Fourier coefficients

The Walsh transform is a Fourier transform for the space of pseudo-Boolean functions in which a pseudo-Boolean function is represented as a linear combination of basis functions called *Walsh functions* [40]. For each subset S of $[n]$, the Walsh function corresponding to S, $\psi_S : \{0,1\}^n \to \mathbb{R}$, is defined as

$$\psi_S(x) = (-1)^{\sum_{i \in S} x_i}$$

for $x \in \{0,1\}^n$. If we define an inner product of two pseudo-Boolean functions f and g as

$$\langle f, g \rangle = \sum_{x \in \{0,1\}^n} \frac{f(x)g(x)}{2^n},$$

the set of Walsh functions, $\{\psi_S \mid S \subseteq [n]\}$, becomes an orthonormal basis of the space of pseudo-Boolean functions. Hence, a pseudo-Boolean function f can be represented as

$$f = \sum_{S \subseteq [n]} \widehat{f}(S)\psi_S,$$

where $\widehat{f}(S) = \langle f, \psi_S \rangle$ is called the *Fourier coefficient* corresponding to S.

Now, we prove Corollary 1.4. Suppose that f is a 2-bounded function satisfying the condition in Corollary 1.4. Then, the Fourier coefficients $\widehat{f}(S)$ of f corresponding to S with $|S| \geqslant 3$ are 0. Decompose f as follows:

$$f = \widehat{f}(\emptyset) + f_1 + f_2,$$

where

$$f_1 = \sum_{S \subseteq [n]:\ |S|=1} \widehat{f}(S)\psi_S \quad \text{and} \quad f_2 = \sum_{S \subseteq [n]:\ |S|=2} \widehat{f}(S)\psi_S.$$

To find the Fourier coefficients of f, we separately find the Fourier coefficients of f_2 and f_1 and then find the Fourier coefficient $\widehat{f}(\emptyset)$.

First, define G_f as the weighted graph such that (i) $V(G_f) = [n]$, (ii) for each pair e of vertices, $e \in E(G_f)$ if and only if $\widehat{f}(e) \neq 0$, and (iii) $w_{G_f}(e) = \widehat{f}(e)$ for all $e \in E(G_f)$. Since the weights of edges of G_f are precisely the Fourier coefficients of f_2, consider the problem of finding the weighted graph G_f. For a cross-additive query (S,T), we have

$$\mu_{S,T}(G_f) = \sum_{e=uv:\ u \in S, v \in T} \widehat{f}(e) = \frac{f(0^n) - f(1_S) - f(1_T) + f(1_{S \cup T})}{4},$$

where 0^n is the vector consisting of n zeros and 1_A for a subset $A \subseteq [n]$ represents the vector consisting of 1 in the coordinates in A and 0 in the rest. This means that any cross-additive query for G_f can be answered by 4 function evaluations of f. Since f has at most m non-zero Fourier coefficients, G_f has at most m edges. Hence, by Theorem 1.2, there exists a non-adaptive algorithm to find the edges of G_f in $\mathcal{O}(\frac{m \log n}{\log m})$ function evaluations. For each edge in G_f, its weight can be found by one cross-additive query. Thus, all the weights of edges in G_f can be found in $\mathcal{O}(m)$ function evaluations in a non-adaptive way. Overall, there exists a 2-round algorithm to find the weights of edges in G_f, consequently, the Fourier coefficients of f_2 in $\mathcal{O}(\frac{m \log n}{\log m})$ function evaluations.

Let x_f be the n-dimensional vector whose ith element is the Fourier coefficient $\widehat{f}(\{i\})$. That is, the elements of x_f are precisely the Fourier coefficients of f_1. For $q \in \{0,1\}^n$, the following holds:

$$\chi_q(x_f) = \frac{(f(0^n) - f_2(0^n)) - (f(q) - f_2(q))}{2}.$$

If we regard x_f as the vector representing the weights of n coins, this equation means that any coin weighing for x_f can be simulated by a constant number of function evaluations of f and f_2. Since f_2 is known by the algorithm for finding G_f and the value of $f(0^n)$ is evaluated by the algorithm, only one function evaluation of f, i.e., $f(q)$, is required to know the value of $\chi_q(x_f)$. The function f_1 has at most m non-zero Fourier coefficients and so $|\mathrm{sp}(x_f)| \leqslant m$. Hence, the non-adaptive algorithm in Theorem 1.3 identifies the non-zero elements of x_f in $\mathcal{O}(\frac{m \log n}{\log m})$ function evaluations. The value of each non-zero element of x_f can be found by one coin weighing, i.e., one function evaluation of f. Thus, all the values of non-zero elements of x_f can be found in $\mathcal{O}(m)$ function evaluations in a non-adaptive way. Overall, there exists a 2-round algorithm to find x_f, consequently, the Fourier coefficients of f_1 in $\mathcal{O}(\frac{m \log n}{\log m})$ function evaluations.

Once we obtain f_2 and f_1, we get the value of $\widehat{f}(\emptyset)$ by $\widehat{f}(\emptyset) = f(0^n) - f_2(0^n) - f_1(0^n)$. This can be done without additional function evaluations of f, since the value of $f(0^n)$ is evaluated by the algorithm for finding G_f. Combining the two algorithms for finding G_f and x_f along with this final step, we obtain an algorithm to find the Fourier coefficients of f using $\mathcal{O}(\frac{m \log n}{\log m})$ function evaluations. Since each of the two algorithms is a 2-round algorithm, the combined algorithm is a 4-round algorithm, which proves Corollary 1.4.

7. Concluding remarks

In this paper, we proved the existence of optimal algorithms for the graph finding problem and two related problems.

Our results for weighted graphs are based on the condition that the weights of edges in absolute value (or corresponding ones for the other problems) are bounded above by a polynomial in n and below by an inverse polynomial in n. The essential part of the condition used in our proofs is that the ratio between the minimum and maximum is polynomially bounded above in n. The polynomial ratio is best possible to derive the optimal bounds in our proofs and as yet we have no idea how to obtain optimal bounds for more general conditions.

For finding weighted graphs, we obtained the results under the condition that m is at least a (large) constant power of $\log n$. Getting the results for smaller m is still open for cross-additive queries and so is for the problem of finding Fourier coefficients. (For additive queries, see [8] that appeared in the review process of this paper.) Also, although the proposed algorithms are optimal, finding an explicit construction of optimal algorithms is still an open problem and, we think, an important research direction. Another important issue in practical applications is the time complexity of the algorithms. In this paper, we focused only on the query complexity and did not try to optimize the time complexity. It would be worth trying to find an optimal algorithm with a reasonable time complexity.

A natural extension of the graph finding problem is the hypergraph finding problem, especially when the hypergraph is k-uniform for $k = 3, 4, \ldots$. An algorithm for the problem would be useful to find the Fourier coefficients of certain k-bounded pseudo-Boolean functions as described in Section 6.

References

[1] M. Aigner, Combinatorial Search, Wiley, New York, 1988.
[2] N. Alon, V. Asodi, Learning a hidden subgraph, SIAM Journal on Discrete Mathematics 18 (4) (2005) 697–712.
[3] N. Alon, R. Beigel, S. Kasif, S. Rudich, B. Sudakov, Learning a hidden matching, SIAM Journal on Computing 33 (2) (2004) 487–501.
[4] D. Angluin, J. Chen, Learning a hidden graph using $\mathcal{O}(\log n)$ queries per edge, in: Proceedings of the 17th Annual Conference on Learning Theory (COLT 2004), Banff, Canada, 2004, pp. 210–223.
[5] D. Angluin, J. Chen, Learning a hidden hypergraph, Journal of Machine Learning Research 7 (2006) 2215–2236.
[6] R. Beigel, N. Alon, M.S. Apaydin, L. Fortnow, S. Kasif, An optimal procedure for gap closing in whole genome shotgun sequencing, in: Proceedings of the Fifth Annual International Conference on Computational Molecular Biology (RECOMB 2001), Montreal, Canada, 2001, pp. 22–30.
[7] M. Bouvel, V. Grebinski, G. Kucherov, Combinatorial search on graphs motivated by bioinformatics applications: A brief survey, in: Proceedings of the 31st International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2005), Metz, France, 2005, pp. 16–27.
[8] N.H. Bshouty, H. Mazzawi, Reconstructing weighted graphs with minimal query complexity, in: Proceedings of the 20th International Conference on Algorithmic Learning Theory (ALT 2009), Porto, Portugal, 2009, pp. 97–109.
[9] N.H. Bshouty, C. Tamon, On the Fourier spectrum of monotone functions, Journal of the ACM 43 (4) (1996) 747–770.
[10] H. Chernoff, A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations, Annals of Mathematical Statistics 23 (1952) 493–509.
[11] S.S. Choi, K. Jung, J.H. Kim, Almost tight upper bound for finding Fourier coefficients of k-bounded pseudo-Boolean functions, in: Proceedings of the 21st Annual Conference on Learning Theory (COLT 2008), Helsinki, Finland, 2008, pp. 123–134.
[12] S.S. Choi, K. Jung, B.R. Moon, Lower and upper bounds for linkage discovery, IEEE Trans. on Evolutionary Computation 13 (2) (2009) 201–216.
[13] S.S. Choi, J.H. Kim, Optimal query complexity bounds for finding graphs, in: Proceedings of the 40th ACM Symposium on Theory of Computing (STOC 2008), Victoria, Canada, 2008, pp. 749–758.
[14] P. Erdős, On a lemma of Littlewood and Offord, Bulletin of the American Mathematical Society 51 (1945) 898–902.

[15] C.G. Esseen, On the Kolmogorov–Rogozin inequality for the concentration function, Z. Wahrscheinlichkeitstheorie Verw. Geb. 5 (1966) 210–216.
[16] C.G. Esseen, On the concentration function of a sum of independent random variables, Z. Wahrscheinlichkeitstheorie Verw. Geb. 9 (1968) 290–308.
[17] W. Fontana, P. Stadler, E. Bornberg-Bauer, T. Griesmacher, I. Hofacker, M. Tacker, P. Tarazona, E. Weinberger, P. Schuster, RNA folding and combinatory landscapes, Physical Review E 47 (3) (1993) 2083–2099.
[18] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison–Wesley, Reading, Massachusetts, 1989.
[19] V. Grebinski, On the power of additive combinatorial search model, in: Proceedings of the 4th Annual International Conference on Computing and Combinatorics (COCOON 1998), Taipei, Taiwan, 1998, pp. 194–203.
[20] V. Grebinski, G. Kucherov, Reconstructing a Hamiltonian cycle by querying the graph: Application to DNA physical mapping, Discrete Applied Mathematics 88 (1998) 147–165.
[21] V. Grebinski, G. Kucherov, Optimal reconstruction of graphs under the additive model, Algorithmica 28 (2000) 104–124.
[22] R.B. Heckendorn, A.H. Wright, Efficient linkage discovery by limited probing, Evolutionary Computation 12 (4) (2004) 517–545.
[23] J. Jackson, An efficient membership-query algorithm for learning DNF with respect to the uniform distribution, Journal of Computer and System Sciences 55 (3) (1997) 42–65.
[24] H. Kargupta, B. Park, Gene expression and fast construction of distributed evolutionary representation, Evolutionary Computation 9 (1) (2001) 1–32.
[25] S.A. Kauffman, Adaptation on rugged fitness landscapes, in: D. Stein (Ed.), Lectures in the Sciences of Complexity, Addison–Wesley, Redwood City, 1989, pp. 527–618.
[26] S.A. Kauffman, The Origins of Order: Self-Organization and Selection in Evolution, Oxford University Press, New York, 1993.
[27] L. Le Cam, On the distribution of sums of independent random variables, in: J. Neyman, L. Le Cam (Eds.), Bernoulli, Bayes, Laplace: Anniversary Volume, Proceedings of an International Research Seminar, Statistical Laboratory, University of California, Berkeley, 1963, Springer-Verlag, New York, 1965, pp. 179–202.
[28] B. Lindström, On B_2-sequences of vectors, Journal of Number Theory 4 (1972) 261–265.
[29] B. Lindström, Determining subsets by unramified experiments, in: J.N. Srivastava (Ed.), A Survey of Statistical Designs and Linear Models, North-Holland, Amsterdam, 1975, pp. 407–418.
[30] J.E. Littlewood, A.C. Offord, On the number of real roots of a random algebraic equation. III, Mat. Sbornik 12 (1943) 277–285.
[31] Y. Mansour, Learning Boolean functions via the Fourier transform, in: V. Roychowdhury, K.Y. Siu, A. Orlitsky (Eds.), Theoretical Advances in Neural Computation and Learning, Kluwer Academic, Dordrecht, 1994, pp. 391–424.
[32] H. Mühlenbein, T. Mahnig, FDA– A scalable evolutionary algorithm for the optimization of additively decomposed functions, Evolutionary Computation 7 (1) (1999) 45–68.
[33] M. Pelikan, D.E. Goldberg, E. Cantú-Paz, Linkage problem, distribution estimation, and Bayesian networks, Evolutionary Computation 8 (3) (2000) 311–340.
[34] L. Reyzin, N. Srivastava, Learning and verifying graphs using queries with a focus on edge counting, in: Proceedings of the 18th International Conference on Algorithmic Learning Theory (ALT 2007), Sendai, Japan, 2007, pp. 285–297.
[35] B.A. Rogozin, An estimate for concentration functions, Theory of Probability and Its Applications 6 (1) (1961) 94–97.
[36] B.A. Rogozin, On the increase of dispersion of sums of independent random variables, Theory of Probability and Its Applications 6 (1) (1961) 97–99.
[37] H.S. Shapiro, S. Söderberg, A combinatory detection problem, American Mathematical Monthly 70 (1963) 1066–1070.
[38] M.J. Streeter, Upper bounds on the time and space complexity of optimizing additively separable functions, in: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2004), Seattle, USA, 2004, pp. 186–197.
[39] H. Tettelin, D. Radune, S. Kasif, H. Khouri, S.L. Salzberg, Optimized multiplex PCR: Efficiently closing a whole-genome shotgun sequencing project, Genomics 62 (1999) 500–507.
[40] J.L. Walsh, A closed set of orthogonal functions, American Journal of Mathematics 55 (1923) 5–24.