Coins with arbitrary weights

Noga Alon * Dmitry N. Kozlov †

Abstract

Given a set of m coins out of a collection of coins of k unknown distinct weights, we wish to decide if all the m given coins have the same weight or not using the minimum possible number of weighings in a regular balance beam. Let $m(n,k)$ denote the maximum possible number of coins for which the above problem can be solved in n weighings. It is known that $m(n,2) = n^{(\frac{1}{2}+o(1))n}$. Here we determine the asymptotic behaviour of $m(n,k)$ for larger values of k. Surprisingly it turns out that for all $3 \leq k \leq n+1$, $m(n,k)$ is much smaller than $m(n,2)$ and satisfies $m(n,k) = \Theta(n \log n / \log k)$.

1 Introduction

Coin-weighing problems deal with the determination or estimation of the minimum possible number of weighings in a regular balance beam that enable one to find the required information about the weights of the coins. There are numerous questions of this type, see, e.g., [GN] and its many references. Here we study the following variant of the old puzzles, which we call *the all equal problem.* Given a set of m coins, we wish to decide if all of them have the same weight or not, when various conditions about the weights are known in advance. The case in which the coins are given out of a collection of coins of k unknown distinct weights is of special interest. Let $m(n,k)$ denote the maximum possible number of coins for which this problem can be solved in n weighings. The case $k = 2$ has been considered in [HH], [KV] and [AV]. The authors of [HH] observed that $m(n,2) \geq 2^n$ for every n. Somewhat surprisingly, this is not tight. In [KV] it is proved that $m(n,2) > 2^n$ for all $n > 2$ and that

$$m(n,2) \leq \frac{3^n - 1}{2}(n+1)^{(n+1)/2}.$$

A preliminary version of part of this paper appears in the Proc. of the 37^{th} FOCS, IEEE (1996).

*Department of Mathematics, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv, Israel. Email address: noga@math.tau.ac.il. Research supported in part by a USA Israeli BSF grant and by the Fund for Basic Research administered by the Israel Academy of Sciences.

†Department of Mathematics, Royal Institute of Technology, S-100 44, Stockholm, Sweden. Email address: kozlov@math.kth.se

These upper and lower bounds for $m(n,2)$ are proved in [KV] by establishing a correspondence between the set of vectors in a Hilbert basis of a certain canonical cone and algorithms that solve the all equal problem for a given number of coins in n weighings.

In [AV] it is shown that the above upper bound for $m(n,2)$ is asymptotically tight, that is,

$$m(n,2) = n^{(\frac{1}{2}+o(1))n},$$

where the $o(1)$-term tends to 0 as n tends to infinity.

In this paper we determine the asymptotic behaviour of $m(n,k)$ for larger values of k. Surprisingly, it turns out that for $k \geq 3$, $m(n,k)$ is much smaller than $m(n,2)$. Our main result is the following.

Theorem 1.1 *There are two absolute positive constants c and C such that for every $n+1 \geq k \geq 3$*

$$c\frac{n\log n}{\log k} \leq m(n,k) \leq C\frac{n\log n}{\log k}.$$

It is worth noting that for $k > n+1$, $m(n,k) = n+1$, as we observe in Section 3.

The proof of Theorem 1.1 is probabilistic, and does not supply an explicit weighing algorithm. For the special case $k = 3$ in which the three potential distinct weights are known to form an arithmetic progression we describe an explicit algorithm.

We also consider several related problems. Our basic approach is similar to the one introduced in [KV] and further studied in [AV], and combines combinatorial and linear algebraic tools. This is briefly explained in Section 2. In Section 3 we observe that if no information on the weights of the coins is given in advance, then $m-1$ weighings are both necessary and sufficient for solving the all equal problem for m coins. We also briefly discuss the number of weighings needed to determine the number of distinct weights of the given m coins. In Section 4 we study the all equal problem for coins whose weights are known to lie in a three term arithmetic progression. Theorem 1.1 is proved in Section 5 and the final Section 6 contains some concluding remarks and open problems.
All logarithms throughout the paper are in base 2, unless otherwise specified.

2 The basic approach

For the case $k = 2$ it is shown in [KV] (see also [AV]) that there is a simple correspondence between coin weighing algorithms solving the all equal problem and certain matrices with entries from $\{1,-1,0\}$. In this section we generalize this approach to the case $k \geq 3$.

To describe this correspondence, consider weighing algorithms for the all equal problem for coins chosen out of a collection of k distinct (unknown) weights. Let $W_{m,k}$ denote the set of all real vectors of length m with positive coordinates and with at most k distinct coordinates. The set $W_{m,k}$ represents the set of all possible weight-vectors of our given m coins. Let $\mathcal{A}_{n,m}$ denote the set of all n by m matrices with $\{0,-1,1\}$-entries whose sum of columns is the zero vector. We can associate

each matrix $A = (a_{ij}) \in \mathcal{A}_{n,m}$ with a weighing algorithm as follows. Put $[m] = \{1, \ldots, m\}$ and define, for each i, $1 \leq i \leq n$, two disjoint subsets L_i and R_i of $[m]$ by $L_i = \{j : \ a_{ij} = -1\}$ and $R_i = \{j : \ a_{ij} = 1\}$. Note that since the sum of each row of A is 0, $|L_i| = |R_i|$. Trying to solve the all equal problem for a given set of m coins denoted $\{1, 2, \ldots, m\}$, the weighing algorithm determined by A consists of n weighings. For $1 \leq i \leq n$ the algorithm compares, in step number i, the coins in L_i with those in R_i. If all weighings are balanced the algorithm declares that all coins have the same weight, otherwise, it states that not all weights are equal. It is not difficult to see that a necessary and sufficient condition for the algorithm to solve correctly the all equal problem for coins with up to k distinct weights is that the only solutions of the system $Ax = 0$ which lie in $W_{m,k}$ are constant vectors. To see this, observe that since $|L_i| = |R_i|$ for every i, if not all the weighings are balanced, then certainly not all the coins have the same weight. If, on the other hand, all weighings are balanced, and it is known that the vector of weights of the coins lies in $W_{m,k}$, then the vector of correct weights must lie in $\ker(A) \cap W_{m,k}$, showing that the algorithm is correct iff there are no nonconstant vectors in $\ker(A) \cap W_{m,k}$.

We have thus seen that any matrix $A \in \mathcal{A}_{n,m}$ defines a weighing algorithm. The converse is also true. Given an optimal weighing algorithm that solves the all equal problem for m coins with up to k weights note, first, that we may assume that the algorithm always compares sets of coins of equal cardinalities. (This is the case because all weights may well be close enough to each other so that the result of any weighing of sets of nonequal cardinalities will be known in advance. See [KV] for the detailed explanation.) We can now define a matrix $A = (a_{ij}) \in \mathcal{A}_{n,m}$ from the algorithm as follows. For each $1 \leq i \leq n$, let $L_i \subset [m]$ and $R_i \subset [m]$ be the two sets of coins the algorithm compares in step number i assuming all previous weighings are balanced. Define $a_{ij} = -1$ if $j \in L_i$, $a_{i,j} = 1$ if $j \in R_i$ and $a_{ij} = 0$ otherwise. Clearly $A \in \mathcal{A}_{n,m}$. The algorithm reports that all coins are of the same weight iff all weighings are balanced, and it is correct for coins with up to k distinct weights iff $\ker(A) \cap W_{m,k}$ consists only of constant vectors.

3 Arbitrary weights

Let us first observe that even if there are no conditions on the weights of the coins at all, one simple algorithm for solving the all equal problem always exists. Namely, one can compare all the coins to a fixed coin, one by one. This will certainly decide whether all the m coins have the same weight or not in $m - 1$ weighings. In the next proposition we observe that if there are no conditions on the weights this number of weighings cannot be improved.

Proposition 3.1 *The most efficient algorithm solving the all equal problem for m coins with arbitrary weights uses $m - 1$ weighings.*

Proof. An algorithm using $m - 1$ weighings is the obvious one described above. Simply pick a fixed coin and compare it to every other coin, each one in its turn. Clearly all the coins have the

same weight iff all weighings are balanced.

To prove that one cannot do better, assume $n < m - 1$ weighings suffice and let $A \in \mathcal{A}_{n,m}$ be the matrix corresponding to an optimal algorithm. Then, if all the weighings are balanced, any vector $w = (w_1, \ldots, w_m) \in \ker(A)$ may be the vector of weights of the m given coins. However, the dimension of $\ker(A)$ is at least $m - n > 1$, showing that it must contain a nonconstant vector, and completing the proof. □

Note that by the above proposition and its proof it follows that for the function $m(n, k)$ defined in the introduction, $m(n, k) = n + 1$ for all $k > n + 1$.

One may be interested in determining how many different weights the coins have. Formulated as a decision problem, this corresponds to the following question: *Given m coins of arbitrary weights, decide whether they have at least k distinct weights or not.* Let $T(m, k)$ denote the minimum number of weighings required to answer this problem.

Proposition 3.2 *For every $m > k > 1$,*

$$\max\{m - 1, m \log_3(k - 1) - k + 1\} \leq T(m, k) \leq m \log_2(2k).$$

Proof. The above problem can be solved by a standard algorithm using at most $m \log_2(2k)$ steps, where in each step we compare two coins. This can be done by a simple binary sorting using the insertion method, while maintaining an ordered list of the distinct weights found so far, as long as their number does not exceed k. On the other hand, it is proved in Björner and Lovász, [BL], using the topological approach introduced there, that any algorithm that performs in each step an arbitrary linear test on the weights of the coins (that is, in each step the algorithm may check if a specified linear form in the weights is positive, negative or zero), and decides in the end if there are at least k distinct weights, must perform, in the worst case, at least $\max\{m - 1, m \log_3(k - 1) - k + 1\}$ tests. Note that the fact that the authors of [BL] allow arbitrary linear tests and not only comparisons between two weights, which might seem a bit artificial in the original context, is essential here. Since every weighing is a special case of a linear test, the desired result follows. □

4 Three weights in arithmetic progression

One of the simplest cases besides that of two weights seems to be the case of three weights, say a, b, c satisfying the simple relation $a + c = 2b$. Let $f(n)$ denote the maximum number m such that it is possible to solve the all equal problem for m coins whose potential weights lie in an (unknown) three-terms arithmetic progression using n weighings. It is not difficult to see that we may restrict our attention here too only to algorithms that compare sets of coins of equal cardinalities in each weighing. Therefore, by the discussion in Section 2, every algorithm corresponds to a matrix $A \in \mathcal{A}_{n,m}$. The algorithm is correct, if and only if the only vectors in $\ker(A)$ whose coordinates all

lie in some three-terms arithmetic progression with positive terms are the constant vectors. This enables one to prove the following simple lemma.

Lemma 4.1 *A matrix $A \in \mathcal{A}_{n,m}$ corresponds to a correct algorithm for solving the all equal problem for m coins whose weights lie in a three term arithmetic progression if and only if the only vectors x with $\{0,-1,1\}$-coordinates in the kernel of A are the constant vectors. This is equivalent to the condition that no two distinct nonempty subsets of the columns of A have the same sum.*

Proof. Suppose there is a nonconstant vector $x = (x_1, \ldots, x_m)$ with $\{0,-1,1\}$-coordinates that lies in $\ker(A)$, and let J denote the all 1 vector of length m. Note that J lies in $\ker(A)$, as the sum of columns of A is the zero vector. Let a, b, c be three distinct positive reals satisfying $a + c = 2b$. Then the vector $y = (b-a)x + bJ$ is in $\ker(A)$, it is not a constant vector, and all its coordinates lie in $\{a, b, c\}$. Hence, A cannot correspond to a correct algorithm.

Conversely, if there is no nonconstant vector x as above, we claim that A corresponds to a correct algorithm. To see this, assume the algorithm is incorrect. Then there is some three-terms progression a, b, c and a nonconstant vector y with coordinates in $\{a, b, c\}$ which lies in $\ker(A)$. This, however, implies that $x = (y - bJ)/(b-a) \in \ker(A)$, and x is clearly a nonconstant vector with $\{0,-1,1\}$-coordinates, contradicting the assumption.

The existence of a nonconstant vector x with $\{0,-1,1\}-$coordinates in $\ker(A)$ is clearly equivalent either to the existence of a proper subset of the columns whose sum is the zero vector, which is equal to the sum of all columns, or to the existence of two disjoint subsets of columns of equal sums. On the other hand, if there are any two nonempty distinct sets of columns with the same sum, then by omitting the columns in their intersection we get two disjoint sets of columns with the above property (and if one of them is empty, then the sum of columns in the other one is equal to the sum of all columns). This completes the proof. □

Remark. Note that the proof actually shows that even if the three potential weights are known in advance, then every correct algorithm that always compares sets of equal cardinalities must correspond to a matrix satisfying the conditions in the lemma. On the other hand, any such matrix corresponds to a correct algorithm, even if the possible weights are not known, and it is only known they lie in a three-terms progression.

Corollary 4.2 *If $m = f(n)$, then*

$$2^m - 1 \leq (2\lfloor m/2 \rfloor + 1)^n \leq (m+1)^n.$$

Therefore,

$$f(n) \leq (1 + o(1)) n \log_2 n,$$

where the $o(1)$-term tends to zero as n tends to infinity.

Proof. Let $A \in \mathcal{A}_{n,m}$ be the matrix corresponding to an optimal algorithm. By Lemma 4.1 all the $2^m - 1$ sums of nonempty subsets of the set of columns of A are distinct. Since the sum of all columns is zero, all such sums must lie in the box $[\ -\lfloor m/2 \rfloor, \lfloor m/2 \rfloor\ ]^n$ and are vectors of integers. Since all of them are distinct, the assertion of the lemma follows. □

Remark. The $(1 + o(1))$-term in the above estimate can be improved to $(\frac{1}{2} + o(1))$ using the second moment method (see, e.g., [AS], Chapter 4.) Here is an outline of the argument. Given a matrix $A = (a_{ij}) \in \mathcal{A}_{n,m}$ corresponding to an optimal algorithm, let $\epsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_m)$ be a random vector in which each ϵ_j is chosen, randomly and independently, to be either 1 or -1, both choices being equally likely. Let X be the random variable $\|A\epsilon\|^2 = \sum_{i=1}^{n} (\sum_{j=1}^{m} a_{ij}\epsilon_j)^2$. By linearity of expectation, the expected value of X satisfies

$$E(X) = \sum_{i=1}^{n} (\sum_{j=1}^{m} a_{ij}^2 + 2 \sum_{1 \le j < j' \le m} a_{ij}a_{ij'}E(\epsilon_j \epsilon_{j'})) = \sum_{i=1}^{n} (\sum_{j=1}^{m} a_{ij}^2) \le nm.$$

Therefore, by Markov's Inequality, the probability that X is at most $2mn$ is at least a half. This means that there are at least 2^{m-1} vectors ϵ as above for which $\|A\epsilon\|^2 \le 2mn$. By the Cauchy-Schwarz Inequality the sum of absolute values of the coordinates of each such vector $A\epsilon$ is at most $n\sqrt{2m}$. Since each such vector is an integral vector with even coefficients, there are only $(O(1)m)^{n/2}$ possible vectors of this type and as Lemma 4.1 implies that all these vectors must be distinct we conclude that $2^{m-1} \le (O(1)m)^{n/2}$, implying that $m \le (\frac{1}{2} + o(1))n \log_2 n$, as claimed.

We next show, by an explicit construction, that $f(n) \ge \Omega(n \log n)$. For every $i \ge 1$, define

$$n_i = \frac{7 \cdot 4^{i-1} - 4}{3}, \qquad m_i = \frac{7}{48} i 4^i + \frac{47}{144} 4^i + \frac{1}{9}.$$

It is easy to check that $n_1 = 1$, $m_1 = 2$, and that $n_{i+1} = 4n_i + 4$ and $m_{i+1} = 4m_i + n_i + 1$ for all $i \ge 1$. Note that for large i, $m_i = (\frac{1}{4} + o(1))n_i \log_4 n_i$.

Lemma 4.3 *For every $i \ge 1$, there is a matrix M_i with entries from $\{-1, 1, 0\}$, having n_i rows and m_i columns, so that the sum of entries of each row of M_i is 0, and the only linear combinations of columns of M_i with coefficients in $\{-1, 1, 0\}$ which vanish are the combinations in which all coefficients are equal.*

Proof. We apply induction on i, starting with the matrix $M_1 = (1, -1)$. Suppose we have already constructed an n_i by m_i matrix $M = M_i$ satisfying the requirements of the lemma. Define a matrix M' with $n_{i+1} = 4n_i + 4$ rows and $m_{i+1} = 4m_i + n_i + 1$ columns as follows. Let O denote an n_i by n_i matrix of zeros, let I denote the identity matrix of order n_i, let j denote a column vector of n_i ones

and let o denote a column vector of n_i zeros. M' is given in the following equation.

$$M' = \begin{bmatrix} M & M & M & M & I & -j \\ M & -M & M & -M & O & o \\ M & M & -M & -M & O & o \\ M & -M & -M & M & O & o \\ 1\,0\ldots0 & 0\,0\ldots0 & 0\,0\ldots0 & 0\,0\ldots0 & 0\,0\ldots0 & -1 \\ 0\,0\ldots0 & 1\,0\ldots0 & 0\,0\ldots0 & 0\,0\ldots0 & 0\,0\ldots0 & -1 \\ 0\,0\ldots0 & 0\,0\ldots0 & 1\,0\ldots0 & 0\,0\ldots0 & 0\,0\ldots0 & -1 \\ 0\,0\ldots0 & 0\,0\ldots0 & 0\,0\ldots0 & 1\,0\ldots0 & 0\,0\ldots0 & -1 \end{bmatrix}$$

It is obvious that M' is a matrix with entries in $\{-1, 1, 0\}$ and that the sum of each of its rows is 0. Put $m = m_i$, $n_i = n$, let $M = (m_{ij})_{1 \le i \le n, 1 \le j \le m}$, and let

$$\mathbf{u} = (x_1, \ldots, x_m, y_1, \ldots, y_m, z_1, \ldots z_m, w_1, \ldots, w_m, t_1, \ldots t_n, r)$$

be a vector with $\{-1, 1, 0\}$-entries for which

$$M'\mathbf{u}^t = 0. \tag{1}$$

To complete the proof we have to show that all coordinates of $\mathbf{u}$ are equal. For each i, $1 \le i \le n$, by summing the rows numbers $i, i+n, i+2n$ and $i+3n$ of the system of equations (1) we conclude that

$$4\sum_{j=1}^{m} m_{ij}x_j + t_i - r = 0.$$

This implies that $t_i - r \equiv 0 \pmod 4$, and since $-2 \le t_i - r \le 2$ it follows that $t_i = r$ for all i and hence that $\sum_{j=1}^{m} m_{ij}x_j = 0$ for all i, implying, by the induction hypothesis, that all the variables x_j are equal.

Similarly, by adding the rows numbers i and $i+2n$ of the system (1) and by subtracting the rows numbers $i+n$ and $i+3n$ we conclude that

$$4\sum_{j=1}^{m} m_{ij}y_j + t_i - r = 0.$$

Since $t_i = r$ for all i we conclude, by the induction hypothesis, that all variables y_j are equal. By a similar argument all the variables z_j are equal and so are all the variables w_j. Moreover, by the last four equations all these variables are equal to r, completing the proof. □

By Lemma 4.3 and Lemma 4.1, and since clearly $f(n)$ is a nondecreasing function of n, we conclude that $f(n) \ge \Omega(n \log n)$. This, together with Corollary 4.2 proves the following.

Theorem 4.4 *There are two absolute positive constants c_1 and c_2 such that*

$$c_1 n \log n \le f(n) \le c_2 n \log n$$

for every n. Moreover, there exists an explicit algorithm for solving the all equal problem for $\lceil c_1 n \log n \rceil$ coins whose potential weights lie in a (known or unknown) three-terms arithmetic progression using n weighings.

5 Three weights or more

In this section we prove Theorem 1.1. Note first that by the discussion in Section 2 $m(n,k)$ is simply the maximum integer m such that there exists an n by m matrix $A \in \mathcal{A}_{n,m}$ for which $\ker(A) \cap W_{m,k}$ contains only constant vectors, where $W_{m,k}$ is the set of all vectors of length m with positive coordinates in which the number of distinct coordinates is at most k. The upper bound in Theorem 1.1 is rather simple, and is proved in the following lemma.

Lemma 5.1 *Suppose $k \geq 3$, put $m = m(n,k)$ and define $r = \lfloor (k-1)/2 \rfloor$. Then*

$$(r+1)^{m-1} \leq (2r(m-1)+1)^n. \tag{2}$$

Therefore,

$$m(n,k) \leq C \frac{n \log n}{\log k}$$

for some absolute constant C.

Proof. Given a matrix $A \in \mathcal{A}_{n,m}$ corresponding to an optimal algorithm, let $v_1, \ldots, v_m$ denote the columns of A. Define

$$S = \Big\{ \sum_{j=2}^{m} r_j v_j, \ r_j \in \mathbf{Z}, 0 \leq r_j \leq r \Big\},$$

where $\mathbf{Z}$ denotes the set of all integers. Then $|S| = (r+1)^{m-1}$. We claim that no two vectors in S are equal. To see this, assume this is false and suppose

$$\sum_{j=2}^{m} r_j v_j = \sum_{j=2}^{m} t_j v_j,$$

where r_j, t_j are integers and there is at least one j for which r_j and t_j differ. Then the vector $y = (0, r_1 - t_1, r_2 - t_2, \ldots, r_m - t_m)$ lies in $\ker(A)$ and is not a constant vector. Since the vector J consisting of m ones is also in $\ker(A)$, so is $(r+1)J + y$, which is in $W_{m,k}$ as each of its coordinates is an integer between 1 and $2r+1 \leq k$. Therefore, A does not correspond to a valid algorithm, showing that indeed all members of S are distinct.

Since each coordinate of any vector in S is an integer whose absolute value cannot exceed $r(m-1)$ the inequality (2) follows, completing the proof. □

Remark. As in the previous section one can apply the second moment method to improve the best estimate obtained for C by the above argument, but since we are not trying to optimize the constants in this section we omit the details.

The lower bound in Theorem 1.1 is proved next, by a probabilistic argument.

Lemma 5.2 *There exists an absolute positive constant c such that for every n and k satisfying $n+1 \geq k \geq 3$,*

$$m(n,k) \geq c\frac{n \log n}{\log k}.$$

Proof. Since $m(n,k) \geq n+1$ for every k the result is trivial for, say, $k \geq n^{1/3}$ (for all $c < 1/3$), and we thus may consider only $k \leq n^{1/3}$. Throughout the proof we assume, whenever this is needed, that n is sufficiently large. To simplify the presentation, we omit all floor and ceiling signs whenever these are not essential. Given a large n, let m be an even integer satisfying $m = (1+o(1))cn \log n/\log k$, where $c < 1/3$ is an absolute positive constant to be chosen later. Let $A \in \mathcal{A}_{n,m}$ be a random matrix obtained by choosing each row of A, randomly and independently, among all row-vectors of length m having exactly half of the coordinates equal to 1 and another half equal to -1. To complete the proof we show that almost surely (that is, with probability that tends to 1 as n tends to infinity) the weighing algorithm corresponding to A solves the all equal problem for coins with up to k distinct weights. To do so, we must show that with high probability there is no nonconstant vector in $W_{m,k}$ that lies in $\ker(A)$. Let $v_j = (v_{1j}, \ldots, v_{nj})^t$, $j = 1, \ldots, m$, denote the columns of A. The existence of a nonconstant vector in $W_{m,k} \cap \ker(A)$ is equivalent to the existence of a partition of $[m] = \{1, 2, \ldots, m\}$ into $l+1 \leq k$ pairwise disjoint nonempty subsets $S_1, S_2, \ldots, S_{l+1}$, such that the vectors $u_i = \sum_{j \in S_i} v_j$ satisfy a linear relation with nonconstant positive coefficients. If there is such a relation, we may take one with the minimum possible value of l. Since $A \in \mathcal{A}_{n,m}$, the sum of the vectors $u_1, \ldots, u_{l+1}$ is zero, and hence such a relation yields a linear relation between any l of the vectors u_i. Without loss of generality we may thus assume that $|S_1| \leq |S_2| \leq \ldots \leq |S_{l+1}|$, that the vectors $u_1, \ldots, u_l$ are linearly dependent and that the vectors $u_1, \ldots, u_{l-1}$ are linearly independent.

For a partition $\mathcal{S} = (S_1, S_2, \ldots, S_{l+1})$ of $[m]$ into pairwise disjoint sets, satisfying $|S_1| \leq |S_2| \leq \ldots \leq |S_{l+1}|$, where $l+1 \leq k \leq n^{1/3}$, put $u_i = \sum_{j \in S_i} v_j$ for $1 \leq i \leq l$, and let $B_{\mathcal{S}}$ denote the event that $u_1, \ldots, u_l$ are linearly dependent, whereas $u_1, \ldots, u_{l-1}$ are linearly independent. By the above discussion, in order to complete the proof, it suffices to prove the following.

Claim: Almost surely, none of the above events $B_{\mathcal{S}}$ occurs.

To prove this claim, fix a partition $\mathcal{S} = (S_1, S_2, \ldots, S_{l+1})$ of $[m]$ as above, put $T = S_l \cup S_{l+1}$, $t = |T|$, and note that $t \geq 2n/k > n^{2/3}$. Let $A_{\mathcal{S}}$ denote the event that there are at least $n/3$ indices i for which

$$\Big|\sum_{j \in T} v_{ij}\Big| > t/10. \tag{3}$$

Note that this event depends only on the choice of the numbers v_{ij} where j lies in the union $S_1 \cup \ldots \cup S_{l-1}$. Using some standard estimates for hypergeometric distributions (or simply the Stirling formula $n! \sim (n/e)^n\sqrt{2\pi n}$), it is not difficult to check that for each fixed i, the probability that (3) holds for i is at most $e^{-\Omega(t)}$. Since the rows of A are chosen independently, this implies that

$$Prob\,[A_{\mathcal{S}}] \leq \binom{n}{n/3} e^{-\Omega(t)n/3} \leq e^{-\Omega(n^{5/3})},$$

where the last inequality follows from the fact that $t > n^{2/3}$.

To estimate the probability $Prob\,[B_{\mathcal{S}}]$ note that

$$Prob\,[B_{\mathcal{S}}] = Prob\,[A_{\mathcal{S}}] \cdot Prob\,[B_{\mathcal{S}} \mid A_{\mathcal{S}}] + Prob\,[\overline{A_{\mathcal{S}}}] \cdot Prob\,[B_{\mathcal{S}} \mid \overline{A_{\mathcal{S}}}]$$

$$\leq Prob\,[A_{\mathcal{S}}] + Prob\,[B_{\mathcal{S}} \mid \overline{A_{\mathcal{S}}}].$$

In order to estimate the conditional probability $Prob\,[B_{\mathcal{S}} \mid \overline{A_{\mathcal{S}}}]$ let us expose, first, all the elements v_{ij} of the matrix A for $1 \leq i \leq n$ and $j \in S_1 \cup \ldots \cup S_{l-1}$ ($= [m] - T$). This enables us to compute $u_1, \ldots, u_{l-1}$, and also supplies the sum in the left hand side of (3) for every i. Since we are interested in bounding the conditional probability above, assume $A_{\mathcal{S}}$ did not happen. If $u_1, \ldots, u_{l-1}$ are not linearly independent, then the event $B_{\mathcal{S}}$ did not happen at all. Otherwise, choose some fixed $l-1$ ($< k$) indices i such that the vectors $u_1, \ldots, u_{l-1}$ restricted to these coordinates are linearly independent. Next, expose all the values v_{ij} for these $l-1$ values of i and for $j \in S_l$. This enables us to compute the unique linear relation between the vectors $u_1, \ldots, u_l$, and hence, if indeed $B_{\mathcal{S}}$ happens, determines uniquely the value of $u_l = \sum_{j \in S_l} v_j$ in each coordinate. There is a set I of at least $2n/3 - (l-1) \geq n/2$ indices i for which (3) does not hold, and for which the values v_{ij} for $j \in S_l$ have not been exposed yet. We now expose them, and estimate the probability that each of these $n/2$ sums $\sum_{j \in S_l} v_{ij}$ for $i \in I$ turns out to be precisely the unique value it has to be in order to satisfy the linear relation which enables the event $B_{\mathcal{S}}$ to occur. It is convenient to consider two cases separately, depending on the size of S_l, which we denote by s. If $|S_l| = s \leq \sqrt{n}$, simply expose, for each fixed $i \in I$, the numbers v_{ij} one by one, and notice that while exposing the last one, the number of positive entries and the number of negative entries in the yet unknown part of the i^{th} row is rather balanced, that is, at least, say, $1/3$ of the entries are negative and at least $1/3$ are positive. Since the last exposed number is uniquely determined, the probability it is the desired number is at most $2/3$. As the rows are chosen independently and $|I| \geq n/2$, we conclude that in this case, the probability that $B_{\mathcal{S}}$ happens is at most $(2/3)^{n/2}$.

If the size s of S_l exceeds $\sqrt{n}$, we note that it surely does not exceed $t/2$ (since $|S_{l+1}| \geq |S_l|$). Therefore, in this case in each row $i \in I$ we are choosing $s \leq t/2$ entries out of a collection of t entries in which the number of -1 and the number of 1 entries do not differ by more than $t/10$. The probability of obtaining any fixed desired number as the sum of the s chosen entries is thus bounded, by some standard estimates for hypergeometric distributions, by $a/\sqrt{s}$, for some absolute constant a. As before, since the rows are independent, in this case the probability $B_{\mathcal{S}}$ occurs is at most $(a/\sqrt{s})^{n/2}$.

Note that in both cases, the above upper bound for the probability is much larger than our $e^{-\Omega(n^{5/3})}$ upper bound for the probability of the event $A_{\mathcal{S}}$. Therefore, we conclude that for each fixed partition $\mathcal{S}$ as above, if s denote the cardinality of S_l, the probability of $B_{\mathcal{S}}$ in case $s \leq \sqrt{n}$ is at most $2(2/3)^{n/2}$, whereas in case $s > \sqrt{n}$, this probability is at most $2(a/\sqrt{s})^{n/2}$.

To complete the proof of the claim observe, now that there are at most

$$\sum_{l=1}^{k-1}\sum_{s=1}^{\sqrt{n}}\binom{m}{ls}(l+1)^{ls}$$

partitions $\mathcal{S}$ with $|S_l| = s \leq \sqrt{n}$. Indeed, once the values of l and s are chosen, the union $S_1 \cup \ldots \cup S_l$ is of size at most ls, and hence we can choose a subset of size ls of $[m]$ (in $\binom{m}{ls}$ possibilities) containing this union. Once this subset is chosen, we can decide for each of its element, in which S_i (including, possibly S_{l+1}) it lies, in at most $(l+1)^{ls}$ possibilities.

A similar argument shows that the number of partitions $\mathcal{S}$ in which the size of S_l satisfies $\sqrt{n} \leq |S_l| \leq m/2k$ is bounded by

$$\sum_{l=1}^{k-1}\sum_{s=\sqrt{n}}^{m/2k}\binom{m}{ls}(l+1)^{ls}.$$

Finally, the number of partitions $\mathcal{S}$ in which the size of S_l exceeds $m/2k$ is bounded by

$$\sum_{l=1}^{k-1}(l+1)^{m},$$

since here we may simply decide, once l is chosen, for each index j to which S_i it belongs.

Combining all the above we conclude that the probability that at least one of the events $B_{\mathcal{S}}$ holds is bounded by $A + B + C$, where

$$A = \sum_{l=1}^{k-1}\sum_{s=1}^{\sqrt{n}}\binom{m}{ls}(l+1)^{ls}2\Big(\frac{2}{3}\Big)^{n/2},$$

$$B = \sum_{l=1}^{k-1}\sum_{s=\sqrt{n}}^{m/2k}\binom{m}{ls}(l+1)^{ls}2\Big(\frac{a}{\sqrt{s}}\Big)^{n/2},$$

and

$$C = \sum_{l=1}^{k-1}(l+1)^{m}2\Big(\frac{a}{\sqrt{(m/2k)}}\Big)^{n/2}.$$

However, since $m = (1+o(1))cn\log n/\log k$ ($< n^2$), and $k \leq n^{1/3}$,

$$A \leq n^{1/3}\sqrt{n}n^{2n^{5/6}}n^{\frac{1}{3}n^{5/6}}\Big(\frac{2}{3}\Big)^{n/2} = o(1).$$

Also,

$$B \leq \sum_{l=1}^{k-1}\sum_{s=\sqrt{n}}^{m/2k}\Big(\frac{2em}{s}\Big)^{ls}2\Big(\frac{a}{\sqrt{s}}\Big)^{n/2}.$$

This is a sum of less than $mk < n^3$ terms, and the logarithm of a typical term is at most

$$ls\log(2em/s) - \frac{n}{8}\log s.$$

However, $ls\log(2em/s)$ is an increasing function of s in the relevant range, as its derivative is $l\log(2em/s) - l \geq l\log(4ek) - l > 0$ and hence $ls\log(2em/s) \leq \frac{m}{2}\log(4ek)$. Since

$$m = (1+o(1))cn\log n/\log k,$$

if c is sufficiently small (say $c < 1/64$), this is smaller than $n\log n/32$. On the other hand $\frac{n}{8}\log s \geq n\log n/16$. This shows that the logarithm of each of the above terms is smaller than $-n\log n/32$ showing that the term itself is at most $n^{-n/32}$, and hence B, which is the sum of less than n^3 such terms, is still $o(1)$.

Finally

$$C \leq n^{1/3}k^m 2\Big(\frac{a}{\sqrt{(m/2k)}}\Big)^{n/2} < n2^{(1+o(1))cn\log n}n^{-n/8} = o(1),$$

where here, again, we applied the fact that c is small, say $c < 1/10$.

Therefore, if $c < 1/64$ then the assertion of the claim holds, completing its proof, and hence completing the proof of the theorem as well. □

6 Concluding remarks

- The results in Section 4 apply to a slightly more general case which we may call *relaxed generic weights*. A set of weights $w_1, \ldots, w_t$ is called *relaxed generic* if any vector of integers $(\lambda_1, \ldots, \lambda_t)$ that satisfies $\sum_{i=1}^t \lambda_i = 0$ and $\sum_{i=1}^t \lambda_i w_i = 0$ is a scalar multiple of the vector $(1, -2, 1, 0, \ldots, 0)$. Note that any set of three terms in arithmetic progression is relaxed generic. Let $f'(n)$ denote the maximum possible number m such that given a set of m coins out of a collection of coins of unknown relaxed generic weights, one can decide if all the coins have the same weight or not using n weighings in a regular balance beam. It is easy to see that the results described in Sections 4 apply to this case (without any essential change in the proofs) and show (constructively) that $f'(n) = \Theta(n\log n)$.

- The techniques described here can be used to study the all equal problem under various similar conditions on the possible weights of the coins. For example, we may assume that the coins are picked out of a collection of coins of weights $w_1, \ldots, w_t$ so that whenever a vector of integers $(\lambda_1, \ldots, \lambda_t)$ satisfies $\sum_{i=1}^t \lambda_i = 0$ and $\sum_{i=1}^t \lambda_i w_i = 0$, it is a scalar multiple of some fixed vector with k nonzero entries. Since most of these variants are somewhat artificial, we do not study them in detail here.

- The proof of the lower bound in Theorem 1.1 described in Section 5 is not constructive. It would be interesting to find a constructive proof yielding an explicit algorithm for the corresponding problem.

References

[Ai] M. Aigner, *Combinatorial Search*, Wiley-Teubner Series in Computer Science, B.G. Teubner and John Wiley and sons, 1988.

[AV] N. Alon and V. H. Vu, *Anti-Hadamard matrices, threshold gates, coin-weighing and indecomposable hypergraphs*, J. Combinatorial Theory, Ser. A, to appear.

[AS] N. Alon and J. H. Spencer, *The probabilistic Method*, Wiley, New York, 1992.

[BL] A. Björner and L. Lovász, *Linear decision trees, subspace arrangements and Möbius functions*, J. AMS 7 (1994), 677–706.

[GN] R. K. Guy and R. J. Nowakowski, *Coin-weighing problems*, Amer. Math. Monthly 102 (1995), 164–167.

[HH] X.D. Hu and F.K. Hwang, *A competitive algorithm for the counterfeit coin problem*, to appear.

[KV] D. N. Kozlov and V. H. Vu, *Coins and cones*, J. Combinatorial Theory, Ser. A, to appear.