# Sometimes Optimal One-Stage Balance Scale Searching with Errors

**Charles Engelhart*** **and Anthony J. Macula***[,1]

Department of Mathematics
SUNY Geneseo
Geneseo, NY 14454

This paper is dedicated to our former and very *balanced* Chair, Donald W. Trasher.

**Abstract.** We specifically give a one-stage and one error-correcting solution to the well-known balance scale problem for a single counterfeit coin. Our method can be generalized to the one-stage multiple error-correcting balance scale problem for a single counterfeit coin. Our solution for the one error-correcting case is optimal in many instances. In particular, for the cases of 12 and 39 coins, our one-stage and one error-correcting solutions use six and seven balance scale comparisons respectively. Using the sphere packing bound, we show that these solutions are optimal.

## 1. Introduction

A survey of counterfeit coin problems is given in [3]. One of the most famous of these can be stated as:

> *Given a set of N coins, all but one of which has uniform weight, what is the best way to find the non-uniform (or counterfeit) coin?*

In this paper, we give a one-stage algorithm for this problem that is as elegant as that given by Dyson in [2]. Moreover, our method can easily be augmented to an error-correcting one-stage algorithm. In a one-stage algorithm, all balance scale *comparisons* (or *weighings*) must be planned in advance. In a multi-stage algorithm, the information gained from (some) initial comparisons is used to determine (some of) the latter comparisons. Our error-free method is similar to that in [1]. Pelc considered the problem of unreliable balance scale comparisons in [5]. However, the method in [5] uses a multi-stage approach and

---

* Both authors were partially supported by NSF-DMS 9973252.

[1] Correspondence, macula@geneseo.edu

Keywords: balance scale search, error-correction, unreliable weighings, counterfeit coin, one-stage algorithm, nonadaptive

Subj. Class.: 05B20, 68P10, 90B40, 94B99

assumes that the non-uniform coin is known at the outset to be heavy. Our one-stage error-correcting algorithm doesn't require any prior knowledge about the non-uniformity of the counterfeit coin.

We assume that we have set of N coins. The goal is to find a one-stage algorithm that identifies the counterfeit and decides whether it is lighter or heavier than the others are. We use the acronym *OB(N)* to denote the problem for N coins. An *OB(N) algorithm* is a one-stage algorithm that identifies the state of the coins. FIGURE 1 is compact representation of our solution to OB(12), the classic problem of 12 coins. In Section 2, we show how to decode this representation.

$$M_3 = \begin{bmatrix} 000 & 111 & 222 & 0 & 1 & 2 \\ 012 & 012 & 012 & 1 & 2 & 0 \\ 120 & 120 & 120 & 1 & 2 & 0 \end{bmatrix}.$$

FIGURE 1

OB(N) is an example of a search problem. If one thinks about a search algorithm as a battery of tests (e.g., balance scale comparisons), then no test in a one-stage search algorithm can be modified by the results of other tests. Thus the order in which the tests are performed is irrelevant. Every parallel algorithm is a one-stage algorithm. In general, one-stage algorithms are desirable when memory or data storage capability is limited. Multi-stage search algorithms are dynamic procedures. They are not predetermined in as far as data from previous stages is used to define the current stage of the search. An example of a multi-stage algorithm is the process of dividing the coins in thirds and then, *depending* upon which third contains the counterfeit coin, repeating this process on *that* third until the counterfeit is isolated.

## 2. The algorithm

We assume that all genuine coins weigh one ounce and that exactly one coin is counterfeit[2]. To see how the matrix $M_3$ in FIGURE 1 is a solution to OB(12), identify the 12 coins with the columns of $M_3$. $M_3$ is an example of a *search matrix* and throughout this paper we will identify coins with columns of some search matrix. The jth coin for $1 \le j \le 12$ is now the jth column of $M_3$. The three rows of $M_3$ describe how to weigh the coins. For $1 \le i \le 3$, put the set of coins $L_i = \{j: (M_3)_{ij} = 1\}$ on the left side of the balance scale and the set of coins $R_i = \{j: (M_3)_{ij} = 2\}$ on the right side of the scale. Call this the *ith comparison*. We use bold-faced letters for vectors. We can now define a $3 \times 1$

[2] We do this for simplicity. If there is no counterfeit, then the output vector **o** will be the constant zero vector.

ternary output vector **o** by setting $\mathbf{o}_i = 0$ if the ith comparison is balanced, $\mathbf{o}_i = 1$ if the ith comparison has the left side lower than the right and finally $\mathbf{o}_i = 2$ if the ith comparison has the left side higher than the right.

Because of the properties of $M_3$ (which are described below) only one of two things can happen. Either **o** is a column of $M_3$ or -**o** mod 3 is a column of $M_3$. If **o** is a column of $M_3$, then the coin that doesn't weigh one ounce (i.e., the counterfeit) is represented by **o** and it is heavier than the others are. If -**o** is a column of $M_3$, then the coin that doesn't weigh one ounce is represented by -**o** and it is lighter than the others are. See Example 1.

**Example 1.** Consider OB(12). Using $M_3$ and the comparison procedure given above, if the seventh coin is lighter, then $\mathbf{o} = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix}$. Observe that **o** is not a column of $M_3$, but $-\mathbf{o} = \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix}$ is. It's the seventh column of $M_3$. Thus the $3 \times 12$ matrix $M_3$ tells us that OB(12) can be solved with three comparisons. From Theorem 1 below, $M_3$ is an optimal solution to OB(12) because it is an optimal *inverse-free balanced matrix*.

**Definition 1.** We say that a non-empty matrix M is an inverse-free balanced (IFB) matrix if:

(a.) M is ternary and each row has the same number of 1s as it has 2s, i.e., every row is *balanced*.
(b.) If **j** *is* a column of M then -**j** mod 3 *isn't* a column of M.
(c.) Each column of M is distinct.
(d.) M doesn't contain a column of zeros.

To solve the general OB(N) in n comparisons, we exhibit an IFB $n \times N$ matrix M and we identify the N coins with the N columns of M. Then for $1 \leq i \leq n$, the ith comparison is:

*Put the set of coins $L_i = \{j\colon M_{ij} = 1\}$ on the left side of the balance scale and the set of coins $R_i = \{j\colon M_{ij} = 2\}$ on the right side.*

The outcomes of these comparisons define an $n \times 1$ ternary output vector **o** by setting $\mathbf{o}_i = 0$ if the ith comparison is balanced, $\mathbf{o}_i = 1$ if the ith comparison has the left side lower than the right, and $\mathbf{o}_i = 2$ if the ith comparison has the left side higher than the right. Because M is IFB, then exclusively **o** or -**o** mod 3 is a column of M. If **o** is a column of M, then the coin that doesn't weigh one ounce is represented by **o** and it is heavier. If -**o** is a column of M, then the coin that doesn't weigh one ounce is represented by -**o** and it is lighter.

## 3. Why the algorithm works

Suppose the jth coin is the counterfeit. If it's heavier, then every comparison in which j is on the left (right) [neither] side of the balance scale results in the left side being lower than (higher than) [level with] the right and the output is recorded as a 1 (2) [0]. However, our matrix M puts j on the left (right) [neither] side of the scale in the ith comparison exactly when $M_{ij} = 1$ (2) [0]. This implies that the jth column of M is equal to **o**.

On the other hand, suppose j is lighter, then every comparison in which j is on the left (right) [neither] side of the balance scale results in the left side being higher than (lower than) [level with] the right and the output is recorded as a 2 (1) [0]. However, our matrix M puts j on the left (right) [neither] side of the scale in the ith comparison exactly when $M_{ij} = 1$ (2) [0]. This implies that the jth column of $M_n$ is equal to -**o** mod 3.

## 4. Constructing maximal IFB matrices

Let $\mu(n) = \max\{N\text{: M is an IFB } n \times N \text{ matrix}\}$. If M is an $n \times \mu(n)$ IFB matrix, then we say that M is a maximal (or optimal) IFB matrix. Clearly an inverse-free ternary matrix[3] with n rows has at most $\frac{3^n - 1}{2} + 1$ columns. From here is straightforward to verify that $\mu(n) = \frac{3^n - 3}{2}$. One might use this observation to try to construct a maximal IFB by starting with the $n \times (3^n - 3)$ matrix that has all possible distinct and non-constant ternary n-sequences as columns. Then from this matrix one could take and keep any column **j** , delete -**j** and simply repeat this procedure until no columns are left. However, this method is slow, tedious, and it doesn't always work. Below we give a recursive method of constructing maximal IFB matrices.

**Definition 2.** Let $B_2 = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \end{bmatrix}$. For n > 2, we define the $n \times 3^{n-1}$ IFB matrix $B_n$ recursively in terms of $B_{n-1}$. For i = 0, 1, 2 and n > 2, let $B_{n-1}(i)$ be the $n \times 3^{n-2}$ matrix derived from $B_{n-1}$ by top row augmenting $B_{n-1}$ with a row of all i's. Then $B_n$ is simply the column augmentation of $B_{n-1}(0)$, $B_{n-1}(1)$ and $B_{n-1}(2)$. We use $B_n$ to construct our solution to OB(N). See FIGURE 2.

---

[3] Just condition (b.) in Definition 1.

$$B_2 = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \end{bmatrix}. \qquad B_3 = \begin{bmatrix} \overset{B_2(0)}{000} & \overset{B_2(1)}{111} & \overset{B_2(2)}{222} \\ 012 & 012 & 012 \\ 120 & 120 & 120 \end{bmatrix}.$$

$$B_4 = \begin{bmatrix} \overset{B_3(0)}{000000000} & \overset{B_3(1)}{111111111} & \overset{B_3(2)}{222222222} \\ 000111222 & 000111222 & 000111222 \\ 012012012 & 012012012 & 012012012 \\ 120120120 & 120120120 & 120120120 \end{bmatrix}.$$

FIGURE 2

**Definition 3.** Let $M_2 = B_2$. For $n > 2$, we inductively define the $n \times \frac{3^n-3}{2}$ matrix $M_n$ as the matrix described in FIGURE 3.

$$\left[\begin{array}{c:c:c:c} B_{n-1}(0) & B_{n-1}(1) & B_{n-1}(2) & \begin{array}{c} M_{n-1} \\ \hdashline \underbrace{120 \quad 120 \quad 120}_{\frac{3^{n-2}-3}{2}\text{times}} \end{array} \end{array}\right]$$

FIGURE 3

It follows from induction that $M_n$ is an IFB matrix. Since it has $\frac{3^n-3}{2}$ columns, then it is a maximal IFB matrix. In Theorem 1 below, we show that $M_n$ is an optimal solution to $OB\left(\frac{3^n-3}{2}\right)$. $M_3$ and $M_4$ are depicted in FIGURES 4 and 5 respectively.

$$M_3 = \left[\begin{array}{c:c:c:c} B_2(0) & B_2(1) & B_2(2) & \begin{array}{c} M_2 \\ \hdashline 120 \end{array} \end{array}\right] = \begin{bmatrix} 000 & 111 & 222 & 0 & 1 & 2 \\ 012 & 012 & 012 & 1 & 2 & 0 \\ 120 & 120 & 120 & 1 & 2 & 0 \end{bmatrix}.$$

FIGURE 4

$$M_4 = \left[\begin{array}{c:c:c:c} B_3(0) & B_3(1) & B_3(2) & \begin{array}{c} M_3 \\ \hdashline 120\ 120\ 120\ 120 \end{array} \end{array}\right] =$$

$$\left[\begin{array}{c:c:c:cccc} 000000000 & 111111111 & 222222222 & 000 & 111 & 222 & 0\ 1\ 2 \\ 000111222 & 000111222 & 000111222 & 012 & 012 & 012 & 1\ 2\ 0 \\ 012012012 & 012012012 & 012012012 & 120 & 120 & 120 & 1\ 2\ 0 \\ 120120120 & 120120120 & 120120120 & 120 & 120 & 120 & 1\ 2\ 0 \end{array}\right].$$

FIGURE 5

## 5. Optimal general solutions for OB(N)

Essentially by definition, every solution to OB(N) has an IFB matrix representation. So to find a solution for OB(N), we need to find an IFB matrix with N columns. If $N = \frac{3^n-3}{2}$, then $M_n$ is an optimal solution because it is a maximal IFB. Suppose that we have N coins where N is a multiple of three and $\frac{3^{n-1}-3}{2} < N \leq \frac{3^n-3}{2}$. We proceed to recursively define an $n \times N$ IFB submatrix, $M_n(N)$, of $M_n$.

Suppose $n = 3$. The possible values of N are 3, 6, 9, and 12. Then one can directly verify that the submatrices of $M_3$ depicted with boldfaced entries respectively in FIGURES 6 (a), (b), (c), and (d) are examples of IFB submatrices $M_3(N)$ with N = 3, 6, 9 and 12.

$$\begin{bmatrix} 000 & 111 & 222 & \mathbf{0} & \mathbf{1} & \mathbf{2} \\ 012 & 012 & 012 & \mathbf{1} & \mathbf{2} & \mathbf{0} \\ 120 & 120 & 120 & \mathbf{1} & \mathbf{2} & \mathbf{0} \end{bmatrix} \qquad \begin{bmatrix} 000 & \mathbf{111} & \mathbf{222} & 0 & 1 & 2 \\ 012 & \mathbf{012} & \mathbf{012} & 1 & 2 & 0 \\ 120 & \mathbf{120} & \mathbf{120} & 1 & 2 & 0 \end{bmatrix}$$

(a) (b)

$$\begin{bmatrix} \mathbf{000} & \mathbf{111} & \mathbf{222} & 0 & 1 & 2 \\ \mathbf{012} & \mathbf{012} & \mathbf{012} & 1 & 2 & 0 \\ \mathbf{120} & \mathbf{120} & \mathbf{120} & 1 & 2 & 0 \end{bmatrix} \qquad \begin{bmatrix} \mathbf{000} & \mathbf{111} & \mathbf{222} & \mathbf{0} & \mathbf{1} & \mathbf{2} \\ \mathbf{012} & \mathbf{012} & \mathbf{012} & \mathbf{1} & \mathbf{2} & \mathbf{0} \\ \mathbf{120} & \mathbf{120} & \mathbf{120} & \mathbf{1} & \mathbf{2} & \mathbf{0} \end{bmatrix}$$

(c) (d)

FIGURE 6

Now suppose that $n > 3$, $\frac{3^n-3}{2} < N \le \frac{3^{n+1}-3}{2}$ and we have a $M_{n-1}(y)$ for any y which is multiple of three and $\frac{3^{n-1}-3}{2} < y \le \frac{3^n-3}{2}$. Then $N = k_{n-1} \cdot 3^{n-1} + \sum_{i=1}^{n-2} k_i 3^i + k_0 3$ where $1 \le k_i \le 3$ for $1 \le i \le n-1$ and $0 \le k_0 \le 1$. Let $y = \sum_{i=1}^{n-2} k_i 3^i + k_0 3$. We have three cases. In each case, the submatrix $M_n(N)$ of $M_n$ is depicted by the boldfaced pieces. See FIGURES 7, 8, and 9.

If $k_{n-1} = 1$, then let

$$M_n(N) = \left[\begin{array}{c:c:c:c} \mathbf{B_{n-1}(0)} & B_{n-1}(1) & B_{n-1}(2) & \dfrac{\mathbf{M_{n-1}(y)}}{\underbrace{\mathbf{120\ \ 120\ \ 120}}_{\frac{y}{3}\text{times}}} \end{array}\right].$$

FIGURE 7

If $k_{n-1} = 2$, then let

$$M_n(N) = \left[\begin{array}{c:c:c:c} B_{n-1}(0) & \mathbf{B_{n-1}(1)} & \mathbf{B_{n-1}(2)} & \dfrac{\mathbf{M_{n-1}(y)}}{\underbrace{\mathbf{120\ \ 120\ \ 120}}_{\frac{y}{3}\text{times}}} \end{array}\right].$$

FIGURE 8

If $k_{n-1} = 3$, then let

$$M_n(N) = \left[\begin{array}{c:c:c:c} \mathbf{B_{n-1}(0)} & \mathbf{B_{n-1}(1)} & \mathbf{B_{n-1}(2)} & \dfrac{\mathbf{M_{n-1}(y)}}{\underbrace{\mathbf{120\ \ 120\ \ 120}}_{\frac{y}{3}\text{times}}} \end{array}\right].$$

FIGURE 9

In each case, it is straightforward to verify that $M_n(N)$ is an IFB matrix. We now have the following theorem. Note that if $N = \frac{3^n - 3}{2}$, then $M_n(N) = M_n$.

**Theorem 1.** Let N be a multiple of three. If $\frac{3^{n-1}-3}{2} < N \leq \frac{3^n-3}{2}$, then $M_n(N)$ is an optimal solution to OB(N).

Proof: By definition, an OB(N) solution must have an IFB matrix representation. Thus the shortest solution to OB(N) is an IFB matrix with $\sigma(N) = \min\{m$: M is an IFB $m \times N$ matrix$\}$ rows. Let $\frac{3^{n-1}-3}{2} < N \leq \frac{3^n-3}{2}$ and suppose N is a multiple of three. Since $M_n(N)$ is IFB $n \times N$ matrix and $\mu(n) = \frac{3^n-3}{2}$, we have that $\sigma(N) = n$ and thus $M_n(N)$ is optimal solution to OB(N). Q.E.D.

**Example 2.** FIGURES 4 and 5 are optimal solutions to OB(12) and OB(39) respectively. Suppose $N = 33 = 3(3^2) + 2(3^1)$. Then $n = 4$, $k_{n-1} = 3$, and $y = 6$. Then $M_4(33)$ is the submatrix of $M_4$ formed by the bold faced entries in FIGURE 10.

$$\left[\begin{array}{c:c:c:cccccc}
\mathbf{000000000} & \mathbf{111111111} & \mathbf{222222222} & 000 & \mathbf{111} & \mathbf{222} & 0 & 1 & 2 \\
\mathbf{000111222} & \mathbf{000111222} & \mathbf{000111222} & 012 & \mathbf{012} & \mathbf{012} & 1 & 2 & 0 \\
\mathbf{012012012} & \mathbf{012012012} & \mathbf{012012012} & 120 & \mathbf{120} & \mathbf{120} & 1 & 2 & 0 \\
\mathbf{120120120} & \mathbf{120120120} & \mathbf{120120120} & 120 & \mathbf{120} & \mathbf{120} & 1 & 2 & 0
\end{array}\right]$$

FIGURE 10

**Example 3.** Suppose $N = 51 = 1(3^3) + 2(3^2) + 1(3) + 3$. Then $n = 5$, $k_{n-1} = 1$, and $y = 24$. Then $M_5(51)$ and $M_4(24)$ are depicted by the bold faced entries in FIGURES 11 and 12 respectively

$$\left[\begin{array}{c:c:c:c}
 & & & \mathbf{M_4(24)} \\ \hdashline
\mathbf{B_4(0)} & B_{n-1}(1) & B_{n-1}(2) & \underbrace{\mathbf{120 \quad 120 \quad 120}}_{8\text{ times}}
\end{array}\right]$$

FIGURE 11

$$\left[\begin{array}{c|c|c|cccccc}
000000000 & \mathbf{111111111} & \mathbf{222222222} & 000 & \mathbf{111} & \mathbf{222} & 0 & 1 & 2 \\
000111222 & \mathbf{000111222} & \mathbf{000111222} & 012 & \mathbf{012} & \mathbf{012} & 1 & 2 & 0 \\
012012012 & \mathbf{012012012} & \mathbf{012012012} & 120 & \mathbf{120} & \mathbf{120} & 1 & 2 & 0 \\
120120120 & \mathbf{120120120} & \mathbf{120120120} & 120 & \mathbf{120} & \mathbf{120} & 1 & 2 & 0
\end{array}\right]$$

FIGURE 12

## 6. Error-correcting OB(N)

For simplicity, we explicitly discuss the single error case and we derive results for $N = \frac{3^n - 3^2}{2}, \frac{3^n - 3}{2}$. Because we are using the general theory of linear codes it is straightforward to generalize our method to correct multiple balance scale errors and/or to consider other multiples of three as values of N.

Suppose we have N coins and an IFB matrix M that is used to search for counterfeit coin j. Assume that at most one error can occur in the comparisons. Then the observed output vector $\mathbf{o}$ differs in at most one entry from $\mathbf{j}$ or $-\mathbf{j}$ where $\mathbf{j}$ is the column M that has been identified with coin j. If $\mathbf{j}$ is the only column with this property, then we can find it by searching for the only column of M such that it or its inverse differs from $\mathbf{o}$ in at most one place. To ensure that $\mathbf{j}$ is the only column with this property, then any two distinct columns of $\mathbf{j}_1$ and $\mathbf{j}_2$ of M must have $H(\mathbf{j}_1, \mathbf{j}_2) \geq 3$ and $H(\mathbf{j}_1, -\mathbf{j}_2) \geq 3$ where $H(\mathbf{j}_1, \mathbf{j}_2)$ is the number of entries where $\mathbf{j}_1$ and $\mathbf{j}_2$ are different. $H(\mathbf{j}_1, \mathbf{j}_2)$ is known as the *Hamming distance* between $\mathbf{j}_1$ and $\mathbf{j}_2$. We write $H^+(M) \geq 3$ if $H(\mathbf{j}_1, \mathbf{j}_2) \geq 3$ and $H(\mathbf{j}_1, -\mathbf{j}_2) \geq 3$ for all pairs of columns $(\mathbf{j}_1, \mathbf{j}_2)$ of M.

To have a one-stage solution to OB(N) that can correct at most one error, we need an IFB matrix E with N columns and $H^+(E) \geq 3$. The main idea behind constructing such an IFB matrix E is to use linear algebra to add redundancy rows to the matrices $M_n(N)$. We do this essentially by multiplying by a suitable generator matrix. See [4]. However, before we can extend our methods, we need to ensure that certain linear combinations of the row vectors of our IFB matrices are balanced. To do this we need to define an IFB submatrix of $M_n$ that is nearly all of $M_n$. Recall that a ternary row vector is balanced if and only if it has an equal number of 1s and 2s.

**Definition 4.** Let $L_3 = B_3$. For $n > 3$, we define the $n \times \frac{3^n - 3^2}{2}$ matrix $L_n$ recursively in terms of $B_{n-1}$. We define $L_n$ as the matrix in FIGURE 13.

$$L_n = \left[ \begin{array}{c|c|c|c} B_{n-1}(0) & B_{n-1}(1) & B_{n-1}(2) & \begin{array}{c} B_{n-1} \\ \hline \underbrace{120 \quad 120 \quad 120}_{3^{n-3}\text{ times}} \end{array} \end{array} \right]$$

FIGURE 13

Note that $L_n$ is an IFB submatrix of $M_n$. $L_n$ is all but the last three columns of $M_n$. $L_4$ is depicted in FIGURE 14.

$$\left[ \begin{array}{c|c|c|ccc} 000000000 & 111111111 & 222222222 & 000 & 111 & 222 \\ 000111222 & 000111222 & 000111222 & 012 & 012 & 012 \\ 012012012 & 012012012 & 012012012 & 120 & 120 & 120 \\ 120120120 & 120120120 & 120120120 & 120 & 120 & 120 \end{array} \right].$$

FIGURE 14

It is straightforward to verify that any linear combination of the rows of $B_n$ over GF(3) is either a balanced or constant row vector. From this observation, we have the following trivial result.

**Proposition 1.** Let R be a $r \times n$ ternary matrix whose first column is the constant 1 vector. Then the matrix product $R \cdot L_n$ (over GF(3),) is a balanced matrix.

**Definition 5.** Let R be a $r \times n$ ternary matrix. R is called a *redundancy matrix* if R has the following properties:

(a.) The first column consists entirely of 1s.
(b.) No column of R is a scalar multiple of another column of R over GF(3).
(c.) Each column has at least two non-zero entries.

Our next result comes from the theory of linear codes. The statement is non-standard, but is specifically aimed at the problem at hand.

**Proposition 2.** Let R be a $r \times n$ redundancy matrix, let M be an $n \times N$ ternary matrix and let G be the $(n+r) \times n$ matrix that is the $n \times n$ identity matrix bottom row augmented with R. This is $G(R) = \left[\frac{I_n}{R}\right]$. Then the matrix product $E = G(R) \cdot M$ has $H(E) \geq 3$.

Proof (sketch). We follow the discussion found in [4]. In [4], the codewords are thought of as row vectors, but we think of codewords (e.g., coins) as column vectors. So we need to consider the matrix transpose to apply the well-known results found in [4]. By Theorem 7.6 in [4], a parity check matrix H for the linear code with generator matrix $G(R)^T = [I_n|R^T]$ is $H = [-R|I_r]$. The conditions (b) and (c) in Definition 5 ensure that any two columns of H are linearly independent. Then our result follows by applying Theorem 8.4 in [4]. Q.E.D.

Let $[L_n \mid -L_n]$ be the matrix $L_n$ column augmented with all the inverses of columns of $L_n$. Since $L_n$ is an IFB, we have that $[L_n \mid -L_n]$ is a $n \times (3^n - 3^2)$ matrix. Now we can construct an IFB matrix E with $H^+(E) \geq 3$. Let R be a $r \times n$ redundancy matrix. From Proposition 2, it follows that $G(R) \cdot [L_n \mid -L_n] = [E \mid -E]$ has $H([E \mid -E]) \geq 3$. From this it follows that $H^+(E) \geq 3$. Since $L_n$ is inverse-free, it follows that E is inverse-free. From Proposition 1, we have that E is an IFB. Thus we have the following result.

**Theorem 2.** If R is a $r \times n$ redundancy matrix, then the matrix $G(R) \cdot L_n = E$ is an IFB matrix and has $H^+(E) \geq 3$. Thus E is a one-error correcting solution to $OB(\frac{3^n - 3^2}{2})$.

The immediate question is how many more comparisons are needed when one moves from the error-free OB(N) to the one error-correcting version of OB(N)? Since for a fixed r there are at most $\frac{3^r - 2r - 1}{2}$ columns in a redundancy matrix with r rows, we have:

**Corollary 1.** If $\frac{3^{r-1} - 2(r-1) - 1}{2} < n \leq \frac{3^r - 2r - 1}{2}$, then we can use the result in Theorem 2 to construct an IFB matrix E that is a solution to the one-error correcting version of $OB(\frac{3^n - 3^2}{2})$ using n + r comparisons.

It would be nice to be have a result like that in Theorem 2 with $M_n$ in place of $L_n$. We can get such a result, but it is tedious to carefully delineate. We give a brief description. Let $[M_n \setminus L_n]$ be the submatrix that consists of the last three columns of $M_n$. The reason why we used $L_n$ instead of $M_n$ in Theorem 2 is because $R \cdot [M_n \setminus L_n]$ can have non-balanced rows even if R is a redundancy matrix. However, if any row in $R \cdot [M_n \setminus L_n]$ is not balanced, then it must be constantly 2. Thus if we have a redundancy matrix R and the first column of $R \cdot [M_n \setminus L_n]$ has no entry which is 2, then $G(R) \cdot M_n = E'$ is a one-stage, one-error correcting solution to OB($\frac{3^n - 3}{2}$).

**Example 4.** Here are the one error-correcting solutions of OB(12) and OB(39). Let $R = \begin{bmatrix} 1 & 2 & 1 & 1 & 1 & 1 & 2 & 1 & 0 & 0 \\ 1 & 0 & 1 & 2 & 2 & 2 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 2 & 1 & 0 & 2 & 2 & 1 & 2 \end{bmatrix}$. For $1 \le i \le 10$, let R(i) be the submatrix that consists of the first i columns of R. Then each R(i) is a redundancy matrix. Consider $M_3$ and $M_4$. Since the first column of $R(i) \cdot [M_i \setminus L_i]$ for i = 3, 4 doesn't contain a 2, we form $G(R(i)) \cdot M_i = E_i'$. FIGURES 15 and 16 are respectively the solutions.

$$G(R(3)) \cdot M_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \hline 1 & 2 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 000 & 111 & 222 & 0 & 1 & 2 \\ 012 & 012 & 012 & 1 & 2 & 0 \\ 120 & 120 & 120 & 1 & 2 & 0 \end{bmatrix} = \begin{bmatrix} 000 & 111 & 222 & 0 & 1 & 2 \\ 012 & 012 & 012 & 1 & 2 & 0 \\ 120 & 120 & 120 & 1 & 2 & 0 \\ \hline 111 & 222 & 000 & 0 & 1 & 2 \\ 120 & 201 & 012 & 1 & 0 & 2 \\ 012 & 120 & 201 & 1 & 0 & 2 \end{bmatrix}.$$

FIGURE 15

$$G(R(4))M_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 1 & 2 & 1 & 1 \\ 1 & 0 & 1 & 2 \\ 1 & 1 & 0 & 2 \end{bmatrix} \begin{bmatrix} 000000000 & 111111111 & 222222222 & 000 & 111 & 222 & 0\ 1\ 2 \\ 000111222 & 000111222 & 000111222 & 012 & 012 & 012 & 1\ 2\ 0 \\ 012012012 & 012012012 & 012012012 & 120 & 120 & 120 & 1\ 2\ 0 \\ 120120120 & 120120120 & 120120120 & 120 & 120 & 120 & 1\ 2\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 000000000 & 111111111 & 222222222 & 000 & 111 & 222 & 0\ 1\ 2 \\ 000111222 & 000111222 & 000111222 & 012 & 012 & 012 & 1\ 2\ 0 \\ 012012012 & 012012012 & 012012012 & 120 & 120 & 120 & 1\ 2\ 0 \\ 120120120 & 120120120 & 120120120 & 120 & 120 & 120 & 1\ 2\ 0 \\ \hline 102021210 & 210102021 & 021210102 & 210 & 012 & 120 & 1\ 0\ 2 \\ 222000111 & 000111222 & 111222000 & 000 & 111 & 222 & 0\ 1\ 2 \\ 210021102 & 021102210 & 102210021 & 222 & 000 & 111 & 0\ 1\ 2 \end{bmatrix}.$$

FIGURE 16

From the sphere packing bound, we can see that our one-error correcting one-stage solutions to OB(12) and OB(39) are optimal. (See Theorem 2.16 in [4].) We consider OB(12). The argument for OB(39) is similar. From the sphere packing bound, any ternary code C of length five with $H(C) \geq 3$ can have at most 22 codewords. If we have an IFB matrix E with $H^+(E) \geq 3$, then the code that consists of the columns of $[E|-E]$ has a Hamming distance of at least three. So by the above augment there can't be a $5\times 12$ IFB matrix E with $H^+(E) \geq 3$, because this would imply the existence of ternary code C of length five with $H(C) \geq 3$ and $|C| = 24$. In other words, a one error-correcting solution to OB(12) requires at least six comparisons.

## 7. References


1. M. Aigner, Combinatorial Search, Wiley, (1988)
2. F. J. Dyson, Note 1931--The problem of the pennies, Math. Gaz. 30, 231-234 (1946)
3. R. K. Guy and R. J. Nowakowski, Coin-weighing problems, American Mathematics Monthly, 102, no. 2, 164-167, (1995)
4. R. Hill, A First Course in Coding Theory, Oxford, (1986)
5. A. Pelc, Detecting a counterfeit coin with unreliable weighings, Ars Combinatoria 27,181-192, (1989)