

A Better Lower Bound for the Number of Magic Squares of Order $4k$

Benjawan Tiamkaew, Boonyisa Saethao and Boorapa Singha^{*}

Department of Mathematics and Statistics, Faculty of Science and Technology, Chiang Mai Rajabhat University, Thailand

Received: 12 December 2025, Received in revised form: 30 April 2026, Accepted: 1 May 2026

Available online: 20 May 2026

Abstract

Background and Objectives: Magic squares of order n are $n \times n$ arrays filled with distinct positive integers $1, 2, 3, \dots, n^2$ arranged in such a way that the sums of the integers in each row, column and each of the main diagonals are the same. This number is called the magic constant or magic sum which is equal to $\frac{n(n^2+1)}{2}$. Magic squares have a long history, the earliest recorded information about magic squares comes from China around 2200 BC and is called "Lo Shu". Magic squares were first mentioned in the western world in the works of Theon of Smyrna and were also used to help calculate horoscopes by a 9th-century Arab astrologer. Next, the work of Greek mathematician Moschopoulos in 1300 helped popularize knowledge about magic squares. Today, over 700 years later, teachers still use this knowledge in the classroom to solve problems and practice addition. The mathematical study of magic squares generally involves their construction, classification and enumeration. Although completely general methods for creating all the magic squares of all orders do not exist, there are studies on algorithms that can be used to construct magic squares in some orders. For example, the De la Loubère method (or Siamese method) is a simple algorithm for creating magic squares of any odd order by placing numbers sequentially, moving diagonally up and to the right, wrapping around the grid, and dropping down one cell when hitting a filled cell or boundary, this technique was observed by the French diplomat Simon de la Loubère in Siam (Thailand) in the late 17th century. The Strachey method constructs magic squares of singly even order $n = 4m + 2$ by combining four smaller identical magic squares of order $2m + 1$ together to form one magic square of order $4m + 2$. Then $0, \frac{n^2}{4}, \frac{n^2}{2}, \frac{3n^2}{4}$ is added to each square and finally certain squares are swapped from the top subsquare to the bottom subsquare. Let $M(2k, k)$ represent the set of $2k \times 2k$ matrices in which every element is either 0 or 1, with the sums of each row and column equal to k and let M_{4k} be the set of magic squares of order $4k$. In 2022, Oboudi showed that each matrix $E \in M(2k, k)$ can be used to construct a magic square in M_{4k} . The author also showed that the number of magic squares of order $4k$ is at least $\frac{\binom{2k}{k}^2}{2}$. To prove this result, Oboudi demonstrated a comparison of the cardinality of three sets by showing that $|M_{4k}| \geq |M(2k, k)| \geq |H| \geq \frac{\binom{2k}{k}^2}{2}$, where H

represents the set of some bipartite graphs that are defined in a specific way. However, comparing the cardinality of the three sets above causes the obtained lower bound $\frac{\binom{2k}{k}^2}{2}$ to possibly deviate from the true value of $|M_{4k}|$. To reduce this error, in this paper we use a new procedure that is simpler than Oboudi's technique and can obtain a lower bound of $|M_{4k}|$ that is closer to the true value.

Methodology: From Oboudi's inequality $|M_{4k}| \geq |M(2k, k)| \geq |H| \geq \frac{\binom{2k}{k}^2}{2}$, in this study we will only consider the condition $|M_{4k}| \geq |M(2k, k)|$ and aim at counting the number of matrices in $M(2k, k)$. This procedure will not compare the number of elements in the set $M(2k, k)$ with other sets. Therefore, it reduces the potential for error. We will begin by considering the condition $|M_{4k}| \geq |M(2k, k)|$ again. Some details of such a condition that have not been discussed in Oboudi's work will be presented in this study. We will present two algorithms for constructing matrices in $M(2k, k)$ and prove their correctness. Then we will show that the output matrices from both algorithms will always be different. After that, we will present formulas to count the number of matrices obtained from both algorithms, one of which is the lower bound that Oboudi has presented in 2022. Then we combine the number of matrices obtained from the two algorithms to find a new lower bound for the number of magic squares of order $4k$. **Main Results:** We have presented the conditions under which the inequality $|M_{4k}| \geq |M(2k, k)|$ is true. After that, two algorithms for constructing matrices in $M(2k, k)$ are presented. Both algorithms produce different matrices. To obtain a new lower bound for the number of magic squares of order $4k$, formulas for counting matrices obtained from both algorithms are investigated. The first is $\frac{\binom{2k}{k}^2}{2}$, which is the lower bound of Oboudi, and the second is $\binom{2k}{k} \cdot \sum_{t=1}^{k-1} \left((k!)^{k-t} \prod_{i=0}^{k-t-1} P(2k-t-i, k) \right)$. After combining the two formulas together, a modified lower bound for the number of magic squares of order $4k$ is introduced, which is

$$\frac{\binom{2k}{k}^2}{2} + \binom{2k}{k} \cdot \sum_{t=1}^{k-1} \left((k!)^{k-t} \prod_{i=0}^{k-t-1} P(2k-t-i, k) \right).$$

Conclusions: This research is a study of a lower bound for the number of magic squares of order $4k$ using the concept from Oboudi's work. The result obtained from this study is a lower bound that are closer to the true value of $|M_{4k}|$ than the original values. The new lower bound is obtained by two algorithms of constructing matrices in $M(2k, k)$. We found that the matrices obtained from the two algorithms are different and there are exactly $\frac{\binom{2k}{k}^2}{2}$ matrices obtained from the first algorithm. This number is equal to the lower bound that Oboudi studied in 2022, but here we use a simpler proof. Moreover, we proved that the second algorithm can construct at least $\binom{2k}{k} \cdot \sum_{t=1}^{k-1} \left((k!)^{k-t} \prod_{i=0}^{k-t-1} P(2k-t-i, k) \right)$ matrices. Finally, when the number of matrices obtained by the two

constructions are combined, a new and better lower bound for the number of magic squares of order $4k$ is obtained.

Keywords: magic square; square matrix; enumerating magic squares

*Corresponding author. E-mail: boorapa_sin@cmru.ac.th

Introduction

Magic squares of order n are $n \times n$ square grids or $n \times n$ matrices consisting of the distinct positive integers $1, 2, 3, \dots, n^2$ arranged such that the sums of the numbers in any horizontal, vertical and both diagonal lines are always the same number, known as the magic constant or magic sum. It has long been known that the magic sum for a magic square of order n is calculated by the formula $\frac{n(n^2+1)}{2}$ (Kraitchik, 1942). For example, the following is a magic square of order 3 whose magic sum is $\frac{3(3^2+1)}{2} = 15$:

$$\begin{bmatrix} 2 & 7 & 6 \\ 9 & 5 & 1 \\ 4 & 3 & 8 \end{bmatrix}.$$

The above definition is the most common type of magic squares, which is sometimes called a normal magic square. However, the concept extends to various definitions based on the numbers used and additional properties. For example, let k and r be fixed nonnegative integers. Recently, Singha (Singha, 2020) introduced a special type of 3×3 magic squares which is denoted by $M_3(k, r)$. This set consists of all 3×3 matrices such that all entries are greater than or equal to k with repetition allowed and the sums along rows, columns, and all the entries from the upper left corner to the lower right corner are equal to r . He presented an algorithm for constructing magic squares in $M_3(k, r)$ and gave a formula to count the number of these magic squares. The topic of creating and counting magic squares were also studied in several other papers (Beck & Van Herick, 2011; Benjamin & Brown, 2014; Das, 1991; Kato & Minato, 2020; Triantafyllou, 2022).

For any positive integer k , let M_{4k} be the set of all magic squares of order $4k$ and $M(2k, k)$ be the set of all $2k \times 2k$ matrices whose entries are either 0 or 1 with the sums of each row and column equal to k . In 2022, Oboudi (Oboudi, 2022) showed that, each matrix in $M(2k, k)$ can be used to construct a magic square in M_{4k} . For convenience and clarity, we quote such method as follows.

Theorem 1. (Oboudi, 2022) Let $A_n = [a_{i,j}]$ be a $n \times n$ matrix defined by : $a_{i,j} = (i-1)n + j$. For any $E = [e_{i,j}] \in M(2k, k)$, a magic square $\emptyset(E)$ of order $4k$ can be constructed by swapping the entries of A_{4k} by the following rules:

- (1) If $e_{i,j} = 1$, then substitute $a_{i,j}$ with $a_{n+1-i,j}$ and substitute $a_{i,n+1-j}$ with $a_{n+1-i,n+1-j}$.
- (2) If $e_{i,j} = 0$, then substitute $a_{i,j}$ with $a_{i,n+1-j}$ and substitute $a_{n+1-i,j}$ with $a_{n+1-i,n+1-j}$.

For example, let $k = 1$ and choose $E = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \in M(2,1)$. We follow the rules as follows.

Since $e_{1,1} = 0$, we swap $a_{1,1}$ with $a_{1,4}$ and swap $a_{4,1}$ with $a_{4,4}$ by the rule (2).

Since $e_{1,2} = 1$, we swap $a_{1,2}$ with $a_{4,2}$ and swap $a_{1,3}$ with $a_{4,3}$ by the rule (1).

Since $e_{2,1} = 1$, we swap $a_{2,1}$ with $a_{3,1}$ and swap $a_{2,4}$ with $a_{3,4}$ by the rule (1).

Since $e_{2,2} = 0$, we swap $a_{2,2}$ with $a_{2,3}$ and swap $a_{3,2}$ with $a_{3,3}$ by the rule (2).

By writing the matrix $A_{4k} = A_4$ according to the definition and swapping the entries of A_4 according to the two rules, we obtain the following magic square $\emptyset(E)$ of order 4 whose magic sum is $\frac{4(4^2+1)}{2} = 34$.

$$A_4 = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix} \xrightarrow{\text{swap entries}} \emptyset(E) = \begin{bmatrix} 4 & 14 & 15 & 1 \\ 9 & 7 & 6 & 12 \\ 5 & 11 & 10 & 8 \\ 16 & 2 & 3 & 13 \end{bmatrix}$$

After proving Theorem 1, Oboudi claimed that $|M_{4k}| \geq |M(2k, k)|$ and then compared the size of the set $M(2k, k)$ with the set H containing some bipartite graphs. He found that $|M_{4k}| \geq |M(2k, k)| \geq |H| \geq \frac{\binom{2k}{k}^2}{2}$, which led him to conclude that $\frac{\binom{2k}{k}^2}{2}$ is a lower bound of $|M_{4k}|$.

When we observe how Oboudi's method works, we found that this method relies on multiple comparisons of set sizes. So, each time the comparison is made, each error occurred. Therefore, to reduce errors, we will consider only an inequality $|M_{4k}| \geq |M(2k, k)|$ and we will try to count the number of matrices in the set $M(2k, k)$ as much as possible. The number we count will be a lower bound of $|M_{4k}|$, which is expected to be better than the previous number.

Methodology

In this section, we prepare some materials used throughout this paper, namely, notations, the basic definitions, and preliminary results.

In view of Theorem 1, we know that each matrix $E \in M(2k, k)$ produces a magic square $\phi(E) \in M_{4k}$. But this theorem alone is not enough to conclude that $|M_{4k}| \geq |M(2k, k)|$ since we do not know whether it is possible for different matrices $E_1, E_2, \dots, E_n$ in $M(2k, k)$ to give the same magic square $\phi(E_1) = \phi(E_2) = \dots = \phi(E_n)$ in M_{4k} . Therefore, in order to ensure that the study results are completely accurate, we are required to present the following results, which have not been discussed in Oboudi's work (Oboudi, 2022).

Lemma 1. If E_1 and E_2 are different matrices in $M(2k, k)$, then $\phi(E_1) \neq \phi(E_2)$.

Proof. Suppose that $E_1 = [e_{i,j}], E_2 = [e'_{i,j}] \in M(2k, k)$ such that $E_1 \neq E_2$. Then $e_{i,j} \neq e'_{i,j}$ for some i, j . Without loss of generality, we may assume $e_{i,j} = 1$ and $e'_{i,j} = 0$. Then Theorem 1 implies that, the entry $a_{i,j}$ of A_{4k} is in the $n+1-i$ th row, j th column of $\phi(E_1)$, while the same entry $a_{i,j}$ is in the i th row, $n+1-j$ th column of $\phi(E_2)$. Since $\phi(E_2)$ contains all entries of A_{4k} and these entries are all distinct, we get that the $n+1-i$ th row, j th column of $\phi(E_2)$ is not equal to $a_{i,j}$. Therefore, the $n+1-i$ th row, j th column of $\phi(E_1)$ and $\phi(E_2)$ contain different entries of A_{4k} . Hence, $\phi(E_1) \neq \phi(E_2)$. ■

From the above lemma, if we let $\omega : M(2k, k) \rightarrow M_{4k}$ be defined by $\omega(E) = \phi(E)$, then it is clear that ω is an injective function. Therefore, we can conclude that $|M_{4k}| \geq |M(2k, k)|$.

In what follows, for a matrix $E \in M(2k, k)$, we denote by r_i the i th row of E . For convenience, we may consider r_i as a sequence of two numbers 0 and 1 with length $2k$ for which the sum of all numbers in this sequence is equal to k . In particular, to refer to a row of E without specifying its location, we may simply use the notation r instead of r_i .

Example 1. Let $k = 3$ and consider the following matrix in $M(6, 3)$.

$$E = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Then $r_1 = 1, 1, 1, 0, 0, 0$, $r_2 = 0, 0, 1, 1, 1, 0$, $r_3 = 1, 0, 0, 0, 1, 1$, $r_4 = 0, 1, 1, 1, 0, 0$, $r_5 = 1, 0, 0, 0, 1, 1$ and $r_6 = 0, 1, 0, 1, 0, 1$.

Definition 1. Let $E \in M(2k, k)$ and let r be a row of E . We call r' an opposite row of r if r' is a row of E obtained by changing all numbers in r from 0 to 1 and 1 to 0.

Example 2. Consider the matrix E in Example 1. We see that $r_3 = r_5$ and they are opposite to r_4 , while r_1, r_2 and r_6 have no opposite rows.

Throughout this paper, the notation $\binom{n}{k}$ denotes the number of ways to choose k elements from a set of n distinct elements, without regard to the order in which the elements are selected. The formula is expressed as $\binom{n}{k} = \frac{n!}{(n-k)!k!}$. In addition, the notation $P(n, k)$, where $P(n, k) = \frac{n!}{(n-k)!}$, represents the number of ways to arrange k elements selected from n distinct elements where order matters.

Lemma 2. Let $E \in M(2k, k)$ and let R_{2k} be the set of all rows that can occur in E . Then $|R_{2k}| = \binom{2k}{k}$.

Proof. Suppose that $E \in M(2k, k)$ and let r be a row of E . Since r is a sequence of 0 and 1 with length $2k$ where the sum of all numbers in this sequence is k , there are the same k numbers of 1s and 0s in this sequence. Then, k positions in this sequence are selected from a total of $2k$ positions to be filled with 1s, and any positions not selected are automatically filled with 0s. It follows that there are $\binom{2k}{k}$ possible ways to form a row r of E . Hence $|R_{2k}| = \binom{2k}{k}$. ■

Results

In this section, we first present two algorithms for constructing some matrices in $M(2k, k)$ and give examples of how the algorithms work. After that, we prove the correctness of both algorithms. Finally, we present formulas for counting the output matrices of the algorithms which give a new lower bound for the number of magic squares of order $4k$.

Algorithm A. The following algorithm produces matrices in $M(2k, k)$ which contain k rows of r and k rows of r' for some $r \in R_{2k}$.

Input : A blank matrix of order $2k$.

Step 1 : Choose $r \in R_{2k}$ and select k rows of the matrix, then fill r into these k rows.

Step 2 : For the remaining k rows that were not selected in Step 1, fill each row with r' .

Output : A matrix in $M(2k, k)$ consisting of k rows of r and k rows of r' .

Example 3. Let $k = 4$, a matrix in $M(8, 4)$ can be constructed using algorithm A as follows.

Step 1. We choose $r = 0, 1, 1, 0, 0, 1, 1 \in R_{2k} = R_8$ and select the 1st, 3rd, 4th and 8th row to be filled with r . We get the following matrix.

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ \hline 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ \hline - & - & - & - & - & - & - & - \\ \hline - & - & - & - & - & - & - & - \\ \hline 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Step 2. For the remaining rows, we fill $r' = 1,0,0,1,1,0,0$ into these rows. Then we get the following matrix, which is in $M(8,4)$.

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Theorem 2. The proof of algorithm *A*.

Proof. Suppose that E is a matrix obtained by the algorithm *A*. Since each row of E is either r or r' , it is clear that the sum of each row is equal to the sum of all numbers in r or r' , which is equal to k . Next, let c_j be the j th column of E . Since E contains k rows of r and k rows of r' , we get that c_j contains k numbers of the j th position of r and also contains k numbers of the j th position of r' . If the j th position of r is equal to 1, then the j th position of r' is equal to 0. On the other hand, if the j th position of r is equal to 0, then the j th position of r' is equal to 1. In both cases, the sum of all $2k$ numbers in c_j is equal to k . Finally, since j is arbitrary, we can conclude that the sum of each column of E is equal to k . Therefore, $E \in M(2k, k)$ and the proof is completed. ■

In what follows, for $E \in M(2k, k)$, we let C_0 be the set of all columns of E with 0 at the top and let C_1 be the set of all columns of E with 1 at the top. It is clear that $|C_0| = |C_1| = k$. Now, we are ready to present the second algorithm.

Algorithm B. The following algorithm produces matrices in $M(2k, k)$ which contain t repeating rows at the top, where $1 \leq t \leq k - 1$.

Input : A blank matrix of order $2k$.

Step 1 : Choose $r \in R_{2k}$ and let t be a fixed integer such that $1 \leq t \leq k - 1$. Fill the first t rows of the matrix with a sequence r .

Step 2 : For the remaining $2k - t$ rows, use the following procedure:

2.1 For any column in C_1 , fill k numbers of 1s in these columns, one per column, without filling repeatedly in the same row.

2.2 For any column in C_0 , in the rows filled with 1s from Step 2.1, fill these rows with k numbers of 0s, one per row and spreading them across each column in C_0 .

Repeat steps 2.1 and 2.2 $k - t$ times.

Step 3 : For the remaining positions in each column in C_1 , fill 0s into these positions.

Step 4 : Fill 1s into the remaining positions, which are in the columns in C_0 .

Output : A matrix in $M(2k, k)$ which contains t repeating rows at the top.

To clarify how the above algorithm works, we present an example of constructing a matrix in $M(2k, k)$ using algorithm *B*. This example will also help to understand the proof that will be presented later.

Example 4. Let $k = 4$, then a matrix in $M(8,4)$ can be constructed using algorithm B as follows.

Step 1. Choose $r = 1, 1, 1, 1, 0, 0, 0 \in R_8$ and let t be a fixed integer such that $1 \leq t \leq 3$, here, we choose $t = 2$. After filling r into the first 2 rows, we get the following matrix.

[illegible]

Step 2. For the last 6 rows, since $k - t = 2$, we repeat Steps 2.1 and 2.2 a total of 2 times. To see the steps clearly, we will use $\mathbf{1}'$ and $\bar{\mathbf{1}}$ to represent the number 1 that will be filled in Step 2.1 in the first and second times respectively. The notations $\mathbf{0}'$ and $\bar{\mathbf{0}}$ will be used in a similar way for Step 2.2. Now, for Step 2.1, by filling $\mathbf{1}'$'s in every column in $C_1 = \{c_1, c_2, c_3, c_4\}$, one per column, without filling repeatedly in the same row. Here, we perform this step as follows.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline 1' & - & - & - & - & - & - & - \\ - & - & - & 1' & - & - & - & - \\ - & 1' & - & - & - & - & - & - \\ - & - & 1' & - & - & - & - & - \\ - & - & - & - & - & - & - & - \end{bmatrix}$$

Next, for Step 2.2, since r_4, r_5, r_6 and r_7 have been filled in Step 2.1, these rows must be continued according to Step 2.2. So, we fill 4 number of $0'$'s into these rows, one per row and one per each column in $C_0 = \{c_5, c_6, c_7, c_8\}$.

We get the following matrix.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline 1' & - & - & - & - & - & 0' & - \\ - & - & - & 1' & - & 0' & - & - \\ - & 1' & - & - & 0' & - & - & - \\ - & - & 1' & - & - & - & - & 0' \\ - & - & - & - & - & - & - & - \end{bmatrix}$$

Steps 2.1 and 2.2 must be performed again. Here, we choose r_4, r_6, r_7 and r_8 as the rows to be filled with $\bar{1}$ s. Then $\bar{0}$ s must be filled in these rows as well. And, as before, $\bar{1}$ s must be filled to every column in C_1 and $\bar{0}$ s must be filled to every column in C_0 . Then we have the following matrix.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline 1' & - & - & \bar{1} & - & - & 0' & - \\ - & - & - & 1' & - & 0' & - & - \\ - & 1' & \bar{1} & - & 0' & - & - & - \\ - & \bar{1} & 1' & - & - & - & - & 0' \\ \bar{1} & - & - & - & - & - & - & - \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline 1' & - & - & \bar{1} & \bar{0} & - & 0' & - \\ - & - & - & 1' & - & 0' & - & - \\ - & 1' & \bar{1} & - & 0' & \bar{0} & - & - \\ - & \bar{1} & 1' & - & - & - & \bar{0} & 0' \\ \bar{1} & - & - & - & - & - & - & \bar{0} \end{bmatrix}$$

Step 3 and Step 4. We follow Step 3 and Step 4 by filling 0 s into the remaining positions of all columns in C_1 . After that, we fill 1 s into the remaining positions of all columns in C_0 . Then we get the following matrix which is a matrix in $M(8,4)$.

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1' & 0 & 0 & \bar{1} & \bar{0} & 1 & 0' & 1 \\ 0 & 0 & 0 & 1' & 1 & 0' & 1 & 1 \\ 0 & 1' & \bar{1} & 0 & 0' & \bar{0} & 1 & 1 \\ 0 & \bar{1} & 1' & 0 & 1 & 1 & \bar{0} & 0' \\ \bar{1} & 0 & 0 & 0 & 1 & 1 & 1 & \bar{0} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

Example 5. Let $k = 3$, we can construct a matrix in $M(6,3)$ using algorithm B as follows.

Step 1. Choose $r = 0,0,1,0,1,1 \in R_6$ and let t be a fixed integer such that $1 \leq t \leq 2$, Here, we choose $t = 2$.

We fill the first 2 rows with r , then we get the following matrix.

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ _ & _ & _ & _ & _ & _ \\ _ & _ & _ & _ & _ & _ \\ _ & _ & _ & _ & _ & _ \\ _ & _ & _ & _ & _ & _ \end{bmatrix}$$

Step 2. For the last 4 rows, we follow Steps 2.1 and 2.2 only one time as $k - t = 1$. We fill 1s into every column in $C_1 = \{c_3, c_5, c_6\}$, one per column, without filling repeatedly in the same row. After that, we fill 0s into each column in $C_0 = \{c_1, c_2, c_4\}$ under the rules of Step 2.2. We get the following matrix.

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ _ & _ & _ & _ & 1 & _ \\ _ & _ & _ & _ & _ & 1 \\ _ & _ & _ & _ & _ & _ \\ _ & _ & 1 & _ & _ & _ \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & _ & _ & _ & 1 & _ \\ _ & 0 & _ & _ & _ & 1 \\ _ & _ & _ & _ & _ & _ \\ _ & _ & 1 & 0 & _ & _ \end{bmatrix}$$

Step 3 and Step 4. We fill 0s and 1s into the remaining positions of all columns in C_1 and C_0 , respectively. Then we get the following matrix which is a matrix in $M(6,3)$.

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

We observe that, Step 2.1 of algorithm B can always be done. This because $|C_1| = k$, so we can fill exactly k numbers of 1 s to all columns in C_1 , one per column. Moreover, after Step 1 is completed, we are left with $2k - t$ blank rows at the bottom of the matrix. So, there is still enough space to fill $k - t$ numbers of 1 s, one per row in some of these $2k - t$ rows. Step 2.2 can always be performed as well, by the same reasons.

Now we are ready to prove the correctness of algorithm B .

Theorem 3. The proof of algorithm B .

Proof. Suppose that E is a matrix obtained by algorithm B . We first show that the sum in every row of E is equal to k . It is clear from Step 1 that, the sum in the first t rows is equal to k as each of these rows contains a sequence r which has k entries of 1 s and k entries of 0 s. For the last $2k - t$ rows which are processed according to Steps 2.1 and 2.2, we see that each time Steps 2.1 and 2.2 are executed, if a 1 is filled into any row, then a 0 is filled into that row as well. So, after repeating Steps 2.1 and 2.2 $k - t$ times, if any row contains any number of 1 s, then that row must contain the same number of 0 s, but may not be equal to the number of 1 s (which is equal to 0 s) in other rows. We assume that this row contains p number of 1 s and p number of 0 s. After performing Steps 3 and 4, $k - p$ number of 0 s and $k - p$ number of 1 s will be filled in the remaining position of this row. Therefore, the sum of all numbers in this row is equal to the number of all 1 s appearing in the row, which is $p + (k - p) = k$. However, after performing Steps 2.1 and 2.2 to the last $2k - t$ rows, there may still be some rows that have not been filled with 1 s and 0 s, that is, these rows are still blank. These rows will then continue with Steps 3 and 4, and finally these rows are filled with 0 s in every column in C_1 , and 1 s in every column in C_0 . In other words, these rows are filled with r' , the opposite row of r . Then the sum in each of these rows is equal to the sum of all numbers in r' which is equal to k . Therefore, the first part of the proof is completed.

Next, we will show that the sum in every column of E is equal to k . We first consider all columns in C_1 . After Step 1 is completed, all columns in C_1 are filled with 1 s in the first t rows of each column. We also see that every time Step 2.1 is done, a 1 is filled to each column in C_1 , one per column. Therefore, after repeating Step 2.1 $k - t$ times, each of these column contains $k - t$ number of 1 s in the last $2k - t$ rows of the column. Thus, each column in C_1 contains $t + (k - t) = k$ number of 1 s. Next, after Steps 3 is completed, the remaining position of these columns are filled with 0 s, which does not affect to the sum in the column. Hence, the sum of each column in C_1 is equal to k . To compute the sum in each column in C_0 , we see that after Step 1 is completed, each of these columns contains 0 s in the first t rows of the column. For the same reason that we considered the column sum in C_1 , we can conclude that after repeating Steps 2.1 and 2.2 $k - t$ times, we have $t + (k - t) = k$ numbers of 0 s in each column in C_0 . Next, all k blank positions in each of these column will be filled with 1 s after

Steps 4 is completed. Hence, every column in C_0 contains k numbers of 0 s and k numbers of 1 s. Therefore, the column sums in C_0 is equal to k . It follows that $E \in M(2k, k)$ and that completes the proof. ■

Lemma 3. The output matrices from algorithm A and the output matrices from algorithm B will always be different.

Proof. Let $E = [e_{i,j}]$ be a matrix obtained from algorithm B . Then all rows of E can be divided into three types as follows:

- (1) the first t rows of E which each row contains the same sequence r .
- (2) the last $2k - t$ rows of E which are filled with 1 s in some columns in C_1 , these rows are also filled with 0 s, with the same number as 1 s, in some columns in C_0 .
- (3) the last $2k - t$ rows of E which are still blank after Step 2.1 and 2.2 are completed, but finally these rows are filled with a sequence r' by Steps 3 and 4.

We notice that, while Steps 2.1 and 2.2 are being repeated, if we fill in 1 s to cover all $2k - t$ rows, then the set of rows of type (3) will be empty. We first show that these three types of rows are different. Clearly, the rows of type (1) and (3) are different. Next, suppose that $C_1 = \{c_{j_1}, c_{j_2}, \dots, c_{j_k}\}$ and let the s th row, which contains a sequence $e_{s,1}, e_{s,2}, \dots, e_{s,2k}$, be an arbitrary row of type (2). By Step 2.1, k numbers of 1 s must be filled into $c_{j_1}, c_{j_2}, \dots, c_{j_k}$, one per column, without filling repeatedly in the same row. So, after the s th row has been repeated in this step $k - t$ times, there can be at most $k - t$ number of 1 s in the set $\{e_{s,j_1}, e_{s,j_2}, \dots, e_{s,j_k}\}$. This implies that, after Step 3 is completed, some entries in this set are zeros. In other words, some of the j_1 th, j_2 th, $\dots$, j_k th entries of the s th row are zeros. Since $C_1 = \{c_{j_1}, c_{j_2}, \dots, c_{j_k}\}$, we get that the j_1 th, j_2 th, $\dots$, j_k th entries of r are all equal to 1 , that is $e_{1,j_1} = e_{1,j_2} = \dots = e_{1,j_k} = 1$. Therefore, the s th row is not equal to r as they contain different number at the same position. Thus, we conclude that the rows of type (2) are different from type (1). Moreover, since the s th row is a row of type (2), the set $\{e_{s,j_1}, e_{s,j_2}, \dots, e_{s,j_k}\}$ contains an elements whose value is 1 . Without loss of generality, suppose that $e_{s,j_1} = 1$. Next, since the j_1 th entry of r is $e_{1,j_1} = 1$, this implies that the j_1 th entry of r' is 0 . It follows that the s th row is not equal to r' as the j_1 th entries of them are not equal. Therefore, we deduce that the rows of type (2) are different from type (3). So these three types of rows are all different.

Now, let F be a matrix obtained from algorithm A . It is clear that each row of F is either a sequence r or r' for some $r \in R_{2k}$ whereas the matrix E obtained from algorithm B contains at least two types of rows. The first of which is a row of type (1), which is a sequence r selected from R_{2k} , and the second one is a row of type (2) which is neither r nor r' . Therefore, the output matrices obtained from algorithms A and B are different. ■

From the above lemma, the matrices obtained by algorithms A and B are different, so if we count the number of matrices obtained by each algorithm and add them together, there will be no double counting. Moreover, since $|M_{4k}| \geq |M(2k, k)|$, this sum will lead to a lower bound of $|M_{4k}|$, as will be proved in the following.

Theorem 4. The number of magic square of order $4k$ will have at least

$$\frac{\binom{2k}{k}^2}{2} + \binom{2k}{k} \cdot \sum_{t=1}^{k-1} \left((k!)^{k-t} \prod_{i=0}^{k-t-1} P(2k-t-i, k) \right).$$

Proof. Let E be a matrix obtained from algorithm A . We will count the number of ways that this matrix can be constructed. It can be seen from the algorithm that the number of output matrices depend on the number of ways that we choose $r \in R_{2k}$ and the number of ways in which we can select k rows from a total of $2k$ rows to be filled with r . In total, we have $\binom{2k}{k} \binom{2k}{k} = \binom{2k}{k}^2$ ways to do both of these steps together. In fact, suppose that U is a set of k rows selected from a total of $2k$ rows of E and V is a set of the rest k rows of E . By algorithm A , if we fill all rows in U with r , then r' must be filled in every row in V . In another way, if all rows in V are first selected and filled with r' , then all rows in U will be filled with r . Both of these operations give the same output matrix in $M(2k, k)$. So, to avoid double counting, we must divide $\binom{2k}{k}^2$ by 2. Hence, the number of matrices constructed from algorithm A is $\frac{\binom{2k}{k}^2}{2}$.

Next, we will count the number of matrices obtained from algorithm B . It can be seen from the algorithm that, if we change at least one of the following : a sequence $r \in R_{2k}$, an integer t which is the number of repeating rows in the top of the matrix, or the way we fill 1s and 0s in the last $2k - t$ rows, then we get a different output matrix of the algorithm. Therefore, the number of ways to construct the matrix depends on the number of ways to perform each of these steps. After filling r to the first t rows in Step 1, we continue with Step 2.1. and we suppose that $C_1 = \{c_{j_1}, c_{j_2}, \dots, c_{j_k}\}$. To fill k numbers of 1s into columns in C_1 , we have $2k - t$ ways to fill a 1 to c_{j_1} . By the rule of Step 2.1, these numbers must not be filled in the same row. So, we have $2k - t - 1$ ways to fill a 1 to $c_{j_2}, \dots, 2k - t - (k - 2) = k - t + 2$ ways to fill a 1 to $c_{j_{k-1}}$ and $2k - t - (k - 1) = k - t + 1$ ways to fill a 1 to c_{j_k} . Next, in Step 2.2, we have to fill k numbers of 0s to k rows that have been filled with 1s from Step 2.1 and fill in all k columns in C_0 . So, we have $k!$ ways to complete this step. Next, we repeat Steps 2.1 and 2.2 again. To fill a 1 to each columns in C_1 the second time, we have to avoid the position in the same column that has been filled with a 1 in the previous time. Therefore, to perform Step 2.1 the second time, we have $2k - t - 1$ ways to fill a 1 to c_{j_1} . To fill a 1 to c_{j_2} , if a 1 that has just filled in c_{j_1} is in the different row from a 1 that was filled in c_{j_2} in the first time, then we have $2k - t - 1 - 1 = 2k - t - 2$ ways to fill a 1 to c_{j_2} . On the other hand, if they are

in the same row, then we have $2k - t - 1$ ways to fill a 1 to c_{j_2} . However, since the purpose of this counting is only to find the lower bound, so we may conclude that there are at least $2k - t - 2$ ways to fill a 1 to c_{j_2} . Similarly, there are at least $2k - t - 3$ ways to fill a 1 to c_{j_3} , ..., there are at least $2k - t - (k - 1) = k - t + 1$ ways to fill a 1 to $c_{j_{k-1}}$ and there are at least $2k - t - k = k - t$ ways to fill a 1 to c_{j_k} . Like in the first time that we perform Step 2.2, we have $k!$ ways to complete Step 2.2 the second time. Similarly, each time Steps 2.1 and 2.2 are executed, the number of ways to proceed can be calculated as follows:

$$\begin{aligned} \text{First time : } & (2k - t)(2k - t - 1) \cdots (k - t + 2)(k - t + 1) \cdot k! = \frac{(2k-t)!}{(k-t)!} \cdot k! \\ & = \frac{(2k-t)!}{(2k-t-k)!} \cdot k! = P(2k - t, k)k! \text{ ways.} \\ \text{Second time : } & (2k - t - 1)(2k - t - 2) \cdots (k - t + 1)(k - t) \cdot k! = \frac{(2k-t-1)!}{(k-t-1)!} \cdot k! \\ & = \frac{(2k-t-1)!}{(2k-t-1-k)!} \cdot k! = P(2k - t - 1, k)k! \text{ ways.} \\ \text{Third time : } & (2k - t - 2)(2k - t - 3) \cdots (k - t)(k - t - 1) \cdot k! = \frac{(2k-t-2)!}{(k-t-2)!} \cdot k! \\ & = \frac{(2k-t-2)!}{(2k-t-2-k)!} \cdot k! = P(2k - t - 2, k) \cdot k! \text{ ways.} \\ & \vdots \\ (k - t)^{\text{th}} \text{ time : } & (2k - t - (k - t - 1))(2k - t - (k - t - 1) - 1) \cdots \\ & \cdots (2k - t - (k - t - 1) - (k - 2))(2k - t - (k - t - 1) - (k - 1)) \cdot k! \\ & = (k + 1)(k) \cdots 3 \cdot 2 \cdot k! = \frac{(k+1)!}{1!} \cdot k! = \frac{(k+1)!}{(k+1-k)!} \cdot k! = P(k + 1, k) \cdot k! \text{ ways.} \end{aligned}$$

By multiplying the number of ways from each time Step 2.1 and 2.2 are performed, we get the following:

$$(k!)^{k-t} \prod_{i=0}^{k-t-1} P(2k - t - i, k).$$

Finally, since $1 \leq t \leq k - 1$ and a sequence r can be chosen from R_{2k} in $\binom{2k}{k}$ ways, we conclude that there are at least $\binom{2k}{k} \cdot \sum_{t=1}^{k-1} ((k!)^{k-t} \prod_{i=0}^{k-t-1} P(2k - t - i, k))$ matrices obtained from algorithm B . Since $|M_{4k}| \geq |M(2k, k)|$, when we combine the number of matrices obtained from both algorithms, we can conclude that there are at least $\frac{\binom{2k}{k}^2}{2} + \binom{2k}{k} \cdot \sum_{t=1}^{k-1} ((k!)^{k-t} \prod_{i=0}^{k-t-1} P(2k - t - i, k))$ magic squares of order $4k$. ■

Example 6. By taking $k = 4$ in Theorem 4, we can compute a lower bound for the number of magic squares of order 16 as follows:

$$\begin{aligned} & \frac{\binom{8}{4}^2}{2} + \binom{8}{4} \cdot \sum_{t=1}^3 \left((4!)^{4-t} \prod_{i=0}^{3-t} P(8 - t - i, 4) \right) \\ & = \frac{70^2}{2} + 70((4!)^3 P(7,4)P(6,4)P(5,4) + (4!)^2 P(6,4)P(5,4) + (4!)P(5,4)) \\ & = 2,450 + 70((13,824)(840)(360)(120) + (576)(360)(120) + (24)(120)) \\ & = 2,450 + 70(501,645,312,000 + 24,883,200 + 2,880) \\ & = 2,450 + 35,116,913,865,600 \\ & = 35,116,913,868,050. \end{aligned}$$

Discussion

In this paper we develop Oboudi's work by adding some missing details as presented in Lemma 1. We have also shown that Oboudi's lower bound, which equal $\frac{\binom{2k}{k}^2}{2}$, can be proved more easily than the original proof (Oboudi, 2022), which is shown in the first part of Theorem 4. We then present a better lower bound for the number of magic squares of order $4k$, which is $\frac{\binom{2k}{k}^2}{2} + \binom{2k}{k} \cdot \sum_{t=1}^{k-1} ((k!)^{k-t} \prod_{i=0}^{k-t-1} P(2k-t-i, k))$. However, there are other forms of matrices in $M(2k, k)$ besides the two forms presented in this research. For example, it can be verified that, if $E \in M(2k, k)$, then a matrix obtained by swapping rows of E is still a matrix in $M(2k, k)$. So if we can count the number of new matrices that are generated by this process, or any other process, then we can obtain a better lower bound for $|M_{4k}|$. This will be left as an open problem for further study.

Conclusions

In this research, we presented the algorithms for constructing two types of matrices in $M(2k, k)$. Then, the counting formulas for the number of such matrices were also presented. Finally, by the result that $|M_{4k}| \geq |M(2k, k)|$, when the two counting formulas are combined, the new lower bound for the number of M_{4k} is obtained, which is $\frac{\binom{2k}{k}^2}{2} + \binom{2k}{k} \cdot \sum_{t=1}^{k-1} ((k!)^{k-t} \prod_{i=0}^{k-t-1} P(2k-t-i, k))$.

Acknowledgments

This paper forms part of work by the first and second authors for a BSc supervised by the third author. All authors acknowledge the support of Chiang Mai Rajabhat University.

References

- Beck, M., & Van Herick, A. (2011). Enumeration of 4×4 magic squares. *Mathematics of computation*, 80(273), 617-621.
- Benjamin, A. T., & Brown, E. J. (2014). Challenging magic squares for magicians. *The College Mathematics Journal*, 45(2), 92-100. <https://doi.org/10.4169/college.math.j.45.2.092>
- Das, M. K. (1991). On concentric magic squares and magic squares of order $4n + 2$. *International Journal of Mathematical Education in Science and Technology*, 22(2), 187-201. <https://doi.org/10.1080/0020739910220203>

Kato, G., & Minato, S. I. (2020). Enumeration of associative magic squares of order 7. *Journal of Information Processing*, 28, 903-910. <https://doi.org/10.2197/ipsjjip.28.903>

Kraitchik, M. (1942). *Mathematical Recreations*. New York: Norton.

Oboudi, M. R. (2022). Constructing and Enumerating of Magic Squares. *Bol. Soc. Paran. Mat*, 40, 1-4.
<https://doi.org/10.5269/bspm.46836>

Singha, B. (2020). The Number of a Type of 3×3 Magic Squares and Its Construction Algorithm. *Burapha Science Journal*, 25(2), 636-646.

Triantafyllou, S. A. (2022, October). Magic squares in order $4k + 2$. In *2022 30th National Conference with International Participation (TELECOM)* (pp. 1-4). IEEE.
<https://doi.org/10.1109/telecom56127.2022.10017312>