Applying Linear Algebra in Solving the Lights Out Puzzle through Programming in Sage

Zheng Chen
Natural Sciences Department
Southern University at New Orleans
New Orleans, LA, USA 70126
email: *zchen@suno.edu*

Joe Omojola
Natural Sciences Department
Southern University at New Orleans
New Orleans, LA, USA 70126
email: *jomojola@suno.edu*

Gino P. Loverde
Department of Education
University of New Orleans
New Orleans, LA, USA 70148
email: *gino.loverde@gmail.com*

Abstract—The Lights Out puzzle presents an interesting problem in mathematics, encompassing elements of linear algebra, number theory and programming. This paper will analyze the mathematics associated with a version of the Lights Out puzzle game developed by Tiger Electronics (c.1995) that is most often presented to users. Our goal is to implement Sage using an algorithm in linear algebra to solve a linear system derived from this game. We attained results regarding the solvability and the structure of the solution space to the linear system. Furthermore, we used Sage to investigate other models of Lights Out puzzles in different settings, such as modifying the shape of the game's array and adding more states of display other than 'on' or 'off'.

Keywords- *reduced row-Echelon forms, Sage, modeling*

*****

I. INTRODUCTION

The Lights Out puzzle is a commercially available, hand-held game that was introduced by Tiger Electronics in the mid-1990's ([4]), although variations date back to the 1970's with a game called Merlin. The game, visually perceived as a square array of buttons (5 by 5), is presented to the player as a configuration where some buttons are 'on' (lit) and some are 'off' (not lit). The goal of the game is to press a series of buttons to turn off all of the lights. When a button is pressed, it changes its own state and also states of its direct neighbors (those buttons to the left, right, above, and below the given pressed-button). Once all the lights are off, the game is over and the player wins.

On the surface, this game seems like a simple child's toy, complete with ear-wrenching, beeping sound effects and a portable, convenient size. For this project, we are interested in a strategy to win the game. We translate each action (pressing a button) into a mathematical calculation, and then investigate the reasoning behind the game.

This paper is based on a previous project in a summer undergraduate research experience (SURE), part of this paper was presented by Mr. Gino in Emerging Researchers National (ERN) Conference in 2012 when he was a undergraduate student in our department and he received the 1st position. This paper aims to present a feasible computational project in applied mathematics for college, especially, sophomore and junior students who have prior learning experience in Linear Algebra. We selected the programming software Sage to solve the problems presented from Lights Out Puzzle because Sage is an open source and freely available and it is a good option for students who have no access to Matlab, Maple or Mathematica. The estimated amount of time a college student would be expected to complete this project including learning and applying Sage is three or four weeks. Through this paper, it shows that it has been a good approach for a student to develop his/her interest in mathematics research by investigating the mathematics reasoning behind a game or a modeling, furthermore, a student can be more motivated to learn programming or new technology through problem solving; we also hope to communicate with mathematics professors about how to advise undergraduate students to do research.

II. MATHEMATICAL MODELING OF THE GAME

The Lights Out puzzle creates a very interesting question in mathematics that involves matrices, linear algebra, number theory and programming.

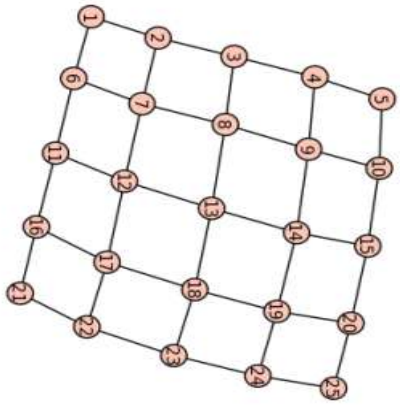

Figure 1

Corresponding to the commercial game board, we first give an ordering to all buttons, for convenience as in Figure 1 (provided through Sage); the ordering can be arbitrary and has no essential impact on solving the problem.

We begin by looking at the 5 by 5 array:

$$M = \begin{pmatrix} 0\,1\,0\,0\,0 \\ 1\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0 \end{pmatrix}$$

and one vector of 25 by 1: $N = (0\,1\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0)^T$, where T is the operation of transposition on a matrix. The matrix M and vector N represent the same configuration in the 5 by 5 board, refer to above Figure 1: there are only two buttons with light on, they are the 2nd and 6th buttons on the board. In this way, each configuration of the board can be represented as a 5 by 5 matrix in a natural way; this matrix can be converted to a vertical vector in the following way: first row then second row and continue. This vertical vector has a size of 25 by 1; it can be converted to a 5 by 5 matrix in "reverse". For example, the above matrices M and N can be converted to each other.

When one button is pressed, there are changes in a few states, which can be represented with a 25 by 1 column vector: we use 1 to indicate a change in the state of the button in the corresponding position, while 0 means the button stays with same state.

For example, when the 1st button is pressed, the vector $Q_1 = (1\,1\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0)^T$.

When the 8th button is pressed, the vector is $Q_8 = (0\,0\,1\,0\,0\,0\,1\,1\,1\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0)^T$.

We call vectors, such as Q_1, Q_8, ***effect vectors***. By collecting all these effect vectors in an increasing order, we can create a 25 by 25 matrix, which is named as a ***Toggle matrix***. A toggle matrix plays a key role in investigating the game of turning off lights; it provides the information of the game rule. Given a starting configuration vector, a toggle matrix will decide whether this configuration is a winning one, that is, all the lights can be turned off after a series of pressing buttons; and the solutions in a winning configuration.

For a 5 by 5 board, it is noticed that the toggle matrix T has a structure as follows in [1]:

$$T = \begin{bmatrix} Z & I & O & O & O \\ I & Z & I & O & O \\ O & I & Z & I & O \\ O & O & I & Z & I \\ O & O & O & I & Z \end{bmatrix},$$ where

$$z = \begin{bmatrix} 1\,1\,0\,0\,0 \\ 1\,1\,1\,0\,0 \\ 0\,1\,1\,1\,0 \\ 0\,0\,1\,1\,1 \\ 0\,0\,0\,1\,1 \end{bmatrix},$$

where I is the 5 by 5 identity matrix, O is the 5 by 5 matrix of all zeros. Notice Z is a symmetric matrix, therefore ***T*** is also symmetric. With these facts in mind, it is much easier for us to input this ***toggle matrix*** in Sage.

Let us look at the game. Each button can only display one of two states (on and off) at a given time. These two states can be represented by 1 and 0 respectively. Furthermore, each action of pressing one button can be represented by a vertical ***effect vector***. We notice a few things: pressing a button twice results in no change; pressing two buttons, the total changes can be expressed in a vector which is sum of two ***effect vectors***, and naturally $1+1=0$ in the summation of components of the vectors; if three or more buttons are pressed, then the order of pressing these buttons has no impact on the total changes of configuration. Furthermore, all linear operations: addition and multiplication, can be conducted in Z_2, the "field of integers modulo 2, 1 for on, and 0 for off", denoted as mod(2) as in [1]. Then a series of button presses result in changes which can be represented as a linear combination of the column vectors in the Toggle matrix in mod(2).

The "turning off all lights" can be modeled as follows: given a starting configuration vector b of 25 by 1, $b = (b_1, b_2, \cdots, b_n)^T$, the Toggle matrix is $T = (T_1, T_2, \cdots, T_{25})$, find a vector p of 25 by 1, such that $b + T * p = 0$ in mod(2) (here $*$ is the multiplication in matrices), which is equivalent to $T * p = b$ in mod(2).

Again, pressing one button an odd number of times makes a change, while pressing one button an even number of times makes no change. Based on this observation, we only need to take into account a series of presses, where no button is pressed more than once.

III. SOLVING THE SYSTEM $T*p=b$ IN MOD(2) IN SAGE

To solve this system, we will program in Sage. The first version of Sage was released in 2005 as a free and open source under the terms of the GNU General Public License, with the initial goals of creating an "open source alternative to Magma, Maple, Mathematica, and MATLAB". The algorithm we use is to use the reduced row-Echelon forms (RREF) of the matrices T and $(T,\ I)$, which will provide critical information we need in deciding whether the system has solutions or no solution.

Recall a matrix is in RREF (in [3]) of a matrix if:

1. All the rows consisting entirely of zeros are at the bottom
2. In each non-zero row, the leftmost non-zero entry is a 1. These are called the leading ones.
3. Each leading one is further to the right than the leading ones of previous rows.
4. The column of each leading one is "clean", that is all other entries in the column are 0.

For matrices with entries in mod(2), we have same concept RREF, and Sage can do the same operations as in the field of all real numbers. Actually, for a prime number n, the ring of integers mod(n) is a field.

The following programming in Sage is to get a matrix G which is the RREF of the toggle matrix T.

```
Z = matrix(IntegerModRing(2),[[1,1,0,0,0],[1,1,1,0,0],
[0,1,1,1,0],[0,0,1,1,1],[0,0,0,1,1]])
I=matrix.identity(IntegerModRing(2), 5);

O=matrix(IntegerModRing(2),5,5,0);

T=block_matrix(5,5,[Z,I,O,O,O,I,Z,I,O,O,O,I,Z,I,O,O,O,I,Z
, I,O,O,O,I,Z]);
G=T.rref()

print G.str()

%md the rank of toggle matrix T in the 5 by 5 lights out game

rT=T.rank()

rT

%md 'the following is the block (T, I), we call it G2, ,G2=(T,
I)'

I2=matrix.identity(IntegerModRing(2), 25);

G2=block_matrix(1,2,[T,I2]);

%md 'Let us denote G2.rref() in Sage as (G,L) or GL '

GL=G2.rref();

%md 'the matrix L is the right half of the matrix GL'

%md 'this matrix L will be used in therorm 1'

L=GL[::, 25::]

show(L)
```

Comments: in this programming, the most important function we use is rref(), which is explained; as to the meaning or syntax of the other functions used in the program, please refer to [6].

This program will produce: $G=\begin{bmatrix} I & g_1 & g_2 \\ 0 & 0 & 0 \end{bmatrix}$, where I is the identity matrix of size 23 by 23, the last two rows of G are zero vectors, g_1 and g_2 are vertical vectors of size 23 by 1, and

$$g_1 = [0\ 1\ 1\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 1\ 1\ ]^T,$$

$$g_2 = [1\ 0\ 1\ \ 0\ 1\ \ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ \ 1\ 0\ 1\ ]^T.$$

It follows from the structure of G that the ***toggle matrix*** T has a rank=23. In order to get the information for solvability or solutions (if they exist), we need two matrices: an ***augmented matrix*** $A=(T,\ b)$ and $G2=(T,\ I)$, where I is identity matrix of 25 by 25. Let us denote G2.rref() in Sage as $(G,\ L)$, then we have $L*T=G$; if $T*p=b$, then $G*p=L*b$; $L*A=L*(T,\ b)=(G,\ L*b)$.

(1) A basic theory in linear algebra tells that the necessary and sufficient condition for the system: $T*p=b$ in mod(2) to have a solutions is that rank(T)=rank(A), or, rank(L*T)=rank($L*T,\ L*b$), or rank(G)=rank($G,\ L*b$).

The above programming in Sage gives the matrix L, which is rather big and we will not display it here. The last two rows of L are:

$$C_1 = [1\ \ 0\ 1\ \ 0\ 1\ \ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ \ 0\ 1\ 1\ 0\ 1\ 0\ 1],$$

$$C_2 = [0\ 1\ \ 1\ 1\ \ 0\ \ 1\ 0\ 1\ 0\ 1\ 1\ 1\ \ 0\ 1\ \ 1\ 1\ 0\ 1\ \ 0\ 1\ 0\ 1\ 1\ 1\ 0].$$

We have $C_1=[g_2^{\,T},\ 0,\ 1]$ and $C_2=[g_1^{\,T},\ 1,\ \ 0]$.

Definition: two vertical vectors v and w are perpendicular if their inner product $v^T*w=0$.

In order that rank(G)=rank($G,\ L*b$)=23, the last two values of $L*b$ must be 0; therefore, we prove the following theorem:

Theorem 1: In the case of a 5 by 5 board, the system: $T*p=b$ in mod(2) has a solution if and only if the vector b is perpendicular to $C_1^{\,T}$ and $C_2^{\,T}$ in mod(2).

Assume $b=(b_1,\ b_2,\cdots,\ b_n)^T$, the above conditions tell that the two sums,

$b_1+b_3+b_5+b_6+b_8+b_{10}+b_{16}+b_{18}+b_{20}+b_{21}+b_{23}+b_{25}$ and

$b_2+b_3+b_4+b_6+b_8+b_{10}+b_{11}+b_{12}+b_{14}+b_{15}+b_{16}+b_{18}+b_{20}+b_{22}+b_{23}+b_{24}$, are even numbers.

By checking carefully all entries in these two sums, we notice that neither of b_7, b_9, b_{17} and b_{19} is in one of the two sums; it follows that the starting state of 7th, 9th, 17th or 19th button has no impact on whether a starting configuration is a winning one. Therefore, we have the following

Corollary 1.1: the values of b_7, b_9, b_{17} and b_{19} have no impact on the solvability of $T*p=b$ in mod(2). In other words, whether a starting configuration is a winning one has nothing to do with the states of the 7th, 9th, 17th and 19th buttons.

Theorem 1 tells that in a winning configuration, two sums

$b_1+b_3+b_5+b_6+b_8+b_{10}+b_{16}+b_{18}+b_{20}+b_{21}+b_{23}+b_{25}$

and

$b_2+b_3+b_4+b_6+b_8+b_{10}+b_{11}+b_{12}+b_{14}+b_{15}+b_{16}+b_{18}+b_{20}+b_{22}+b_{23}+b_{24}$

are even, from which two sates, for example, b_{24} and b_{25}, can be decided by the others. Using the equation $b_1+b_3+b_5+b_6+b_8+b_{10}+b_{16}+b_{18}+b_{20}+b_{21}+b_{23}+b_{25}=0$ in mod(2), we have $b_{25}=b_1+b_3+b_5+b_6+b_8+b_{10}+b_{16}+b_{18}+b_{20}+b_{21}+b_{23}$ in mod(2); using the equation $b_2+b_3+b_4+b_6+b_8+b_{10}+b_{11}+b_{12}+b_{14}+b_{15}+b_{16}+b_{18}+b_{20}+b_{22}+b_{23}+b_{24}=0$ in mod(2), we have $b_{24}=b_2+b_3+b_4+b_6+b_8+b_{10}+b_{11}+b_{12}+b_{14}+b_{15}+b_{16}+b_{18}+b_{20}+b_{22}+b_{23}$ in mod(2).

In the other direction, if we choose the states of 23 buttons arbitrarily except 24th and 25th buttons, and choose states of 24th and 25th buttons as decided as above, then this starting configuration is a winning one by theorem 1. Therefore, we have the following:

Corollary 1.2: There are total 2^{23} winning configurations among all 2^{25} possible starting configurations; the probability to win for a randomly chosen starting configuration is 25%.

Also, we may draw some results about the solutions of the system $T*p=0$.

(1)The Kernel space of the matrix T

We know the kernel space of a matrix T is the space of all vectors p such that $T*p=0$. For this vector p, we have $L*T*p=0$, or $G*p=0$. Assume $p=[p_1,p_2,\cdots,p_{25}]^T$, based on the structure of the matrix G, we know that

$[p_1,p_2,\cdots,p_{23}]^T+p_{24}*g_1+p_{25}*g_2=0$ in mod(2),

therefore, $[p_1,p_2,\cdots,p_{23}]^T=p_{24}*g_1+p_{25}*g_2$ in mod(2),where p_{24} and p_{25} can take the values of 0 and 1. There are two independent solutions:

when $p_{24}=1$ and $p_{25}=0$, the solution is

$[p_1,p_2,\cdots,p_{23},1,0]^T=[g_1^T,1,0]^T=C_2^T$;

when $p_{24}=0$ and $p_{25}=1$, the solution is

$[p_1,p_2,\cdots,p_{23},0,1]^T=[g_2^T,0,1]^T=C_1^T$.

The above expressions give the structure of the kernel space of the matrix T.

Corollary 1.3: the system $T*p=0$ in mod(2) has the following solutions:

$p=[p_1,p_2,\cdots,p_{23},p_{24},p_{25}]^T=p_{25}*C_1^T+p_{24}*C_2^T$,

where p_{24} and p_{25} take values of 0 or 1.

(2) The solution space of $T*p=b$ when the conditions in theorem 1 are satisfied

If $T*p=b$, then $L*T*p=L*b$, that is, $G*p=L*b$; let us denote $L*b=\beta$, and $\beta=(\beta_1,\beta_2,\cdots,\beta_{25})^T$. Notice that $T*p=b$ in mod(2) has a solution iff that $\beta_{24}=\beta_{25}=0$; furthermore we can prove as follows that this β is one solution of $T*p=b$ in nod(2).

Since $G=\begin{bmatrix} I & g_1 & g_2 \\ 0 & 0 & 0 \end{bmatrix}$, we have $G*p=L*b$ is same as

$$\begin{bmatrix} I & g_1 & g_2 \\ 0 & 0 & 0 \end{bmatrix}*[p_1,p_2,\cdots,p_{25}]^T=[\beta_1,\beta_2,\cdots,\beta_{25}]^T,$$

that is,

$$[p_1,p_2,\cdots,p_{23},0,0]^T+p_{24}\begin{bmatrix} g_1 \\ 0 \\ 0 \end{bmatrix}+p_{25}\begin{bmatrix} g_2 \\ 0 \\ 0 \end{bmatrix}=[\beta_1,\beta_2,\cdots,\beta_{23},0,0]^T$$

in mod(2),

$[p_1,p_2,\cdots,p_{23},0,0]^T=[\beta_1,\beta_2,\cdots,\beta_{23},0,0]^T+p_{24}[g_1^T,0,0]^T+p_{25}[g_2^T,0,0]^T$ in mod(2).

Therefore,

$[p_1,p_2,\cdots,p_{23},p_{24},p_{25}]^T=(\beta_1,\beta_2,\cdots,\beta_{23},0,0)^T+p_{24}[g_1^T,1,0]^T+p_{25}[g_2^T,0,1]^T$ in mod(2). Particularly, when $p_{24}=0$ and $p_{25}=0$, then we have one solution

$[p_1,p_2,\cdots,p_{23},0,0]^T=[\beta_1,\beta_2,\cdots,\beta_{23},0,0]^T=\beta$.

We notice that $C_1=[g_1^T,1,0]$ and $C_2=[g_2^T,0,1]$, then we have the following :

Corollary 1.4: If b is a winning starting configuration, then there are 4 winning strategies or 4 solutions to the system $T*p=b$ in mod(2): β, $\beta+C_1^T$, $\beta+C_2^T$ and $\beta+C_1^T+C_2^T$, where $\beta=L*b$.

Now we have obtained the conditions of solvability of the system $T*p=b$ in mod(2) and all the possible

solutions when it is solvable; therefore we have completely figured out the puzzle of the game "lights out".

Example: Consider the following starting configurations:

$S_1 = [1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],$

$S_2 = [1,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0].$

The first one S_1 is not a winning configuration: as to the vector S_1, $b_1 = b_2 = 1,$ $b_1 + b_3 + b_5 + b_6 + b_8 + b_{10} + b_{16} + b_{18} + b_{20} + b_{21} + b_{23} + b_{25} = 1$ in mod(2). While S_2 is a winning configuration: as to the vector S_2, $b_1 = b_2 = b_6 = 1,$ $b_1 + b_3 + b_5 + b_6 + b_8 + b_{10} + b_{16} + b_{18} + b_{20} + b_{21} + b_{23} + b_{25} = b_1 + b_6 = 0$; $b_2 + b_3 + b_4 + b_6 + b_8 + b_{10} + b_{11} + b_{12} + b_{14} + b_{15} + b_{16} + b_{18} + b_{20} + b_{22} + b_{23} + b_{24} = b_2 + b_6 = 0$ in mod(2). The solutions are:

[1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0],

[1 1 1 1 0 1 0 1 0 1 1 1 0 1 1 1 0 1 0 1 0 1 1 1 0],

[0 0 1 0 1 1 0 1 0 1 0 0 0 0 0 1 0 1 0 1 1 0 1 0 1],

[0 1 0 1 1 0 0 0 0 0 1 1 0 1 1 0 0 0 0 0 1 1 0 1 1].

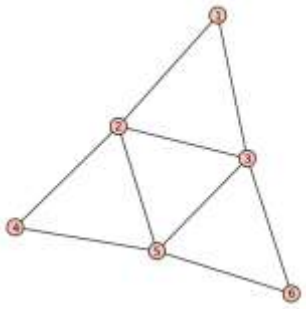

Figure 2: Triangular board

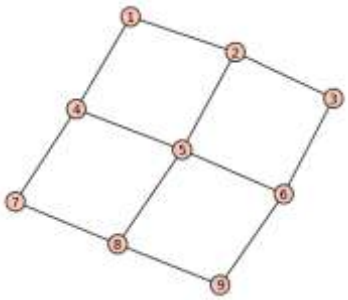

Figure 3: 3 by 3 board

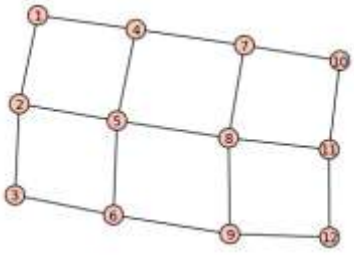

Figure 4: 3 by 4 board

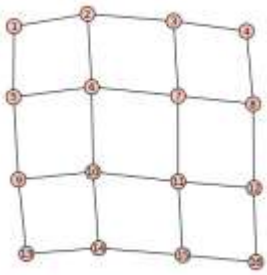

Figure 5: 4 by 4 board

IV. GAME VARIATIONS

Following the rules as the commercial Lights Out puzzle, the game can be played on almost any shaped arrays. We have the boards in Figure 2 to Figure 5.

We discuss a few variations in above figures. They are a triangular board with six buttons, one 3 by 3 board with 9 buttons, one 3 by 4 board with 12 buttons and one 4 by 4 board with 16 buttons. All the buttons are on the grid points. In the triangular board, 3 middle points on three sides are connected. We will use the same game rule: each button has two states: on and off; pressing once on each button will change the states of itself and all neighboring buttons (connected by a side).

Though intuitive sense is always important in mathematics and all sciences, sometimes we cannot trust it too much. By using Sage with same algorithms, we may check the above 4 cases one by one through Sage. In Sage, we only need to input a different Toggle matrix T in each case, then the same programming can be applied to this Toggle matrix T. We then have the following results:

In the cases of Figure 2, Figure 3 and Figure 4, the toggle matrices are of full rank; for any starting configuration b in each case, the system $T * p = b$ in mod(2) are all solvable and it has only one solution. In other words, any starting configuration is a winning one, and there is only one way to win.

But in Figure 5, it is another story. For this case, based on the programming in Sage as we do in the case of a 5 by 5 board, we may draw the following conclusions:

Theorem 2: In the case of 4 by 4 board in Figure 5, the system $G * p = b$ in mod(2) has a solution if and only if the starting vector b is perpendicular to the transposed vectors of the following 4 vectors in mod(2):

[1 0 0 0 1 1 0 0 1 0 1 0 0 1 1 1],

[0 1 0 0 1 1 1 0 0 0 0 1 1 1 0 1],

[0 0 1 0 0 1 1 1 1 0 0 0 1 0 1 1],

[0 0 0 1 0 0 1 1 0 1 0 1 1 1 1 0].

By using the same reasoning as the one for corollary 1.2, we can draw the following corollary.

Corollary 2.1: In Figure 5, there are total 2^{12} winning configurations among all 2^{16} possible starting configurations; the probability to win for a randomly chosen starting configuration is 6.25%.

As to the kernel space and solution space, we can get similar results as in the case of a 5 by 5 board. We omit these similar results; and here we like to introduce a model based on a soccer ball. We know that an official soccer ball is made of leather pieces of 12 regular pentagons (that are usually painted black) and 20 regular hexagons (painted white) in Figure 6.

We assume that on each pentagon or hexagon, a button together with light on or off is installed; pressing once on each button will change the states of itself and all its neighbors. Obviously, the button on a pentagon has 5 neighbors while the button on a hexagon has 6 neighbors. In this case, what we can we tell as to the same problem of turning off lights?

For this model based on a soccer ball, in order to describe a configuration and the changes in states, we need to give an ordering to all buttons from 1 to 32. This can be done in an arbitrary way without essential impact on the solvability and the possible solutions to win. The toggle matrix in this case is of size 32 by 32, which is hard to handle and be reduced by hand to Echelon form. With Sage, we input this 32 by 32 matrix in mod(2) and apply the same programming in Sage. To our surprise, we find that the toggle matrix is of full rank 32 in mod(2), and it follows that each possible starting configuration is a winning one, and there is only one way to win.

V. Three Color States

The other variation here is to include one more state. Instead of two states of on and off, each button displays one of three colors (yellow, green, or purple); we define that a yellow button, once pressed, would change to green; a green button, once pressed, would change to purple; a purple button, once pressed, would change to yellow. We now shift mod(2) to mod(3) in which 0 for yellow, 1 for green, and 2 for purple. Now it takes three presses before a button is brought back to its original state. Therefore, in this case, all operations, vectors, matrices, solving the system are processed in mod(3). Since the toggle matrices remain the same, we can utilize almost same Sage programming in identifying a winning configuration and the strategies to win.

Figure 6: A standard soccer

%md on the triangle form, use A to replace T in section IV

```
A=matrix(IntegerModRing(3),[[1,1,1,0,0,0,],[1,1,1,1,1,0],[1,1,1,0,1,1],[0,1,0,1,1,0],[0,1,1,1,1,1],[0,0,1,0,1,1]])
GA=A.rref()
show(GA)
rA=A.rank()
%md the rank of A in triangle case
rA
```

When work in the state of three colors, we only need to replace IntegerModRing(2) by IntegerModRing(3) in the above programming; The toggle matrix in this case of soccer ball is of size 32 by 32, which is too big to be included here.

We collect key information of the toggle matrices in the Table 1.

Table 1: Comparison of ranks of the toggle matrices

Size of boards	Ranks in mod(2)	Ranks in mod(3)
Triangle	6	6
3 by 3	9	9
3 by 4	12	10
4 by 4	12	14
5 by 5	23	22
Soccer ball	32	32

Based on the ranks and rref(T, I), we can tell more about the solvability, the kernel space and solutions of the systems in the rings mod(3) of integers. We will not include these details.

Acknowledgment

This research was partially supported by a grant from the National Science Foundation (HRD-0928797).

References

[1] Anderson, M. and Feil, T.: Turning Lights Out with Linear Algebra. Mathematics Magazine, 71,(4).300-303, 1998

[2] Kosan, T.: Sage for Newbies, Retrieved from http://sage.math.washington.edu/home/tkosan/newbies_book/sge_for_newbies_v1.23.pdf, 2007

[3] Lay,David C.: Linear Algebra and its Applications, Pearson, 4th edition, 2011

[4] Madsen, M. A. (2010): Lights Out: Solutions Using Linear Algebra. Summation, 3. 36-40, 2010

[5] Martin-Sanchez, O. and Pareja-Flores, C.: Two Reflected Analyses of Lights Out. Mathematics Magazine,74(4). 295-304, 2001

[6] http://www.sagemath.org