Summation
May 2010, pp. 36-40
http://ripon.edu/macs/summation

Lights Out: Solutions Using Linear Algebra

Matthew A. Madsen
Department of Mathematics and Computer Science – Ripon College

***Abstract*—Microcontrollers are seen everywhere in everyday life. Items such as cell phones, remotes, and electronic games use them. This paper investigates the electronic game Lights Out that uses a microcontroller and explains how one could compute a winning strategy for the game. It demonstrates a couple of different methods and works through the proofs of them.**

***Key Words*—Lights Out, Linear Algebra, Microcontroller**

I. Introduction

MICROCONTROLLERS are small computers on a single integrated circuit consisting internally of a relatively simple CPU, clock, timers, I/O ports, and memory. They are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, power tools, and toys. The device that this paper is on is the electronic game Lights Out.

In this paper, I will go over several different methods that can be used to find a solution for the game Lights Out. I will first go through a mathematical method that involves the uses of linear algebra to find a winning strategy using the fewest number of moves. Then I'll talk about a method that doesn't involve the use of an electronic device and one that was described to me by one of my colleagues.

II. Lights Out Rules and Observations

A. Basic Rules

Lights Out is an electronic game manufactured by Tiger Toys in 1995. It consists of 25 buttons, each of which can be lit up, which form a 5 by 5 array. Each button can be in one of two states (on or off). At the start of play, a pattern of lit buttons is chosen by the microcontroller. By pressing the buttons, the player can alter the state of some subset of the buttons according to the rule. The effect of pressing a button is to toggle the state of that button, and its immediate vertical and horizontal neighbors. The goal of the game is to turn all the buttons to the off state.

The game has three modes. In the first mode, you are presented with 50 increasingly difficult puzzles. You have to solve one puzzle before you can move on to the next. The goal of this game is to solve each initial set-up in the minimum possible number of moves; however you are allowed to go over this minimum by ten. The second mode gives you another 1000 puzzles to solve. Mode three allows the player to enter your own puzzles, however not all the possible configurations of states are solvable. This will be demonstrated later in the paper.

B. Observations

The first initial observation about the game is that each button needs to be pressed no more than once. This observation comes from the fact that pressing a button twice is like not pressing it at all. Since pressing a button toggles the state of that button and of its immediate vertical and horizontal neighbors, pressing that same button again will reverse the states and toggle the buttons back to their original states. The key consequence of this observation is that all the calculations can be done in modulo 2. That is, pressing a button an even number of times is like not pressing it at all and pressing it an odd number of times is the same as pressing it only once. This consequence will be very important in calculating the solutions mathematically.

The second initial observation is that the state of each button only depends on how many times it and its immediate vertical and horizontal neighbors are pressed. This observation indicates is that the order in which you press the buttons is irrelevant to the resulting configuration.

A third observation comes directly from the first observation. If we start with the board completely off and press a sequence of buttons to get a configuration, then starting with that configuration and pressing the same sequence of buttons will result in all the lights turning off.

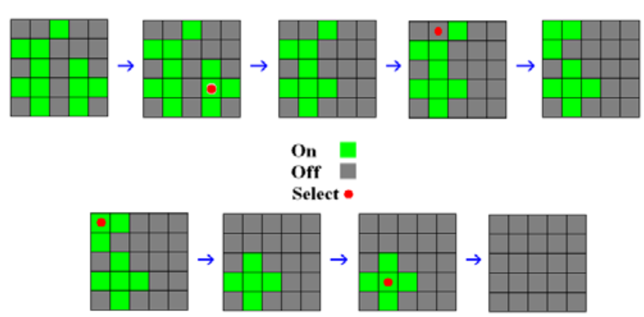

Figure 1. An example of how to play the game. The green squares represent lights that are on. The red dots are the selection of the play. This starting configuration can be solved with only pressing four buttons

Work presented to the college March 23, 2010 as the second talk in a series of three talks about microcontrollers

Matthew A. Madsen currently pursuing his baccalaureate degree in mathematics at Ripon College in Ripon, WI 54971 USA (e-mail:madsenm@ripon.edu).

III. Linear Algebra Solution

One way to guarantee a solution in the fewest number of buttons pressed is to use linear algebra. This can be done with basic matrix operations, some Gauss-Jordan elimination, and an understanding of the column and null space of a matrix.

Since there are two states to the buttons, on and off, we can do all of our calculations in modulo 2 by letting one represent on and zero represent off. We can then think of the 5 by 5 array as a 25 by 1 vector, as in

$$\vec{b} = (b_1, b_2, b_3, \cdots, b_{24}, b_{25})^T \tag{1}$$

where b_1, b_2, b_3, … ,b_{24}, b_{25} will represent the state of each button related to Fig. (2). This vector will be referred to as the configuration of the array. A configuration $\vec{b}$ is obtained by pressing a sequence of buttons, which we will denote as

$$\vec{x} = (x_1, x_2, x_3, \cdots, x_{24}, x_{25})^T \tag{2}$$

where x_1, x_2, x_3, … ,x_{24}, x_{25} will represent whether that particular button is pressed. If we start with all the lights out and the configuration $\vec{b}$ is obtained by strategy $\vec{x}$, then

$$\begin{aligned}
b_1 &= x_1 + x_2 + x_6 \\
b_2 &= x_1 + x_2 + x_3 + x_7 \\
b_3 &= x_2 + x_3 + x_4 + x_8 \\
\vdots &= \quad \vdots \\
b_{13} &= x_8 + x_{12} + x_{13} + x_{14} + x_{18} \\
\vdots &= \quad \vdots \\
b_{24} &= x_{19} + x_{23} + x_{24} + x_{25} \\
b_{25} &= x_{20} + x_{24} + x_{25}.
\end{aligned} \tag{3}$$

It is now straightforward to rewrite this system of linear equations as the matrix product $A\vec{x} = \vec{b}$, where A is the 25 by 25 matrix

$$A = \begin{bmatrix} Z & I & O & O & O \\ I & Z & I & O & O \\ O & I & Z & I & O \\ O & O & I & Z & I \\ O & O & O & I & Z \end{bmatrix}. \tag{4}$$

In this matrix, I is the 5 by 5 identity matrix, O is the 5 by 5 matrix of all zeros, and Z is the matrix

$$Z = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}. \tag{5}$$

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Figure 2. How the buttons on a 5 by 5 are viewed as a 25 by 1 vector

A key observation to make about Z is that it is symmetric, and therefore A is also symmetric. To find a solution for the A given configuration $\vec{b}$, we must find the strategy $\vec{x}$ that satisfies $A\vec{x} = \vec{b}$. To do this, we will prove two theorems.

A. Theorem 1

Given a starting configuration, we first need to check to see if the configuration is winnable. We say that configuration $\vec{b}$ is winnable if there is a strategy $\vec{x}$ that satisfies $A\vec{x} = \vec{b}$.[1] Therefore, Theorem 1 states a configuration $\vec{b}$ is winnable if and only if it is orthogonal to the two vectors $\vec{n}_1$ and $\vec{n}_2$, where

$$\vec{n}_1 = (0,1,1,1,0,1,0,1,0,1,1,1,0,1,1,1,0,1,0,1,0,1,1,0,0)^T \tag{6}$$

and

$$\vec{n}_2 = (1,0,1,0,1,1,0,1,0,1,0,0,0,0,0,1,0,1,0,1,1,0,1,0,1)^T. \tag{7}$$

To prove this theorem, we first need to understand some definitions and theorems about the column and null space of a matrix. For configuration $\vec{b}$ to be winnable, it must have a strategy $\vec{x}$ that satisfies $A\vec{x} = \vec{b}$, meaning that configuration $\vec{b}$ is winnable if and only if it belongs to the column space of A, denoted as Col (A).

Since A is symmetrical, the row space of A, denoted as Row (A), is equal to Col (A). From the definitions of the row space and the null space, we know that Row (A) is equal to the orthogonal complements of the null space of A, denoted Null (A). Since the null space of a matrix is not affected by the Gauss-Jordan elementary row operations, let matrix E be the reduced row echelon form of matrix A. We can then see that Null (A) = Null (E).

It then follows that $\vec{b}$ is winnable if and only if it belongs to the orthogonal complements of Null (E). Since we are looking for the orthogonal complements of Null (E), we need to find an orthogonal basis for the Null (E). To find a basis for the Null (E), we will interpret matrix E as a system of equations and then solve for the dependent variables. We can see in the equations

$$\begin{array}{rcl} 0 & = & e_1 + e_{25} \\ 0 & = & e_2 + e_{24} \\ 0 & = & e_3 + e_{24} + e_{25} \\ \vdots & = & \vdots \\ 0 & = & e_{23} + e_{24} + e_{25} \end{array} \Rightarrow \begin{array}{rcl} e_1 & = & e_{25} \\ e_2 & = & e_{24} \\ e_3 & = & e_{24} + e_{25} \\ \vdots & = & \vdots \\ e_{23} & = & e_{24} + e_{25} \\ e_{24} & = & e_{24} \\ e_{25} & = & e_{25} \end{array}, \tag{8}$$

that by doing the calculations in modulo 2, that the negatives become positive. By writing the independent variables in terms of the dependent variables we get

$$\begin{array}{lcl} e_1 & = & \\ e_2 & = & \\ e_3 & = & \\ \vdots & = e_{24} & \\ e_{23} & = & \\ e_{24} & = & \\ e_{25} & = & \end{array} \begin{bmatrix} 0 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \\ 0 \end{bmatrix} + e_{25} \begin{bmatrix} 1 \\ 0 \\ 1 \\ \vdots \\ 1 \\ 0 \\ 1 \end{bmatrix}. \tag{9}$$

The resulting collection of vectors will be the basis for the Null (E). So a basis for the Null (E) is $\vec{n}_1$ and $\vec{n}_2$, the same as (6) and (7).

To check to see if these two vectors form an orthogonal basis, we must simply show the dot product of the two vectors is equal to zero. We find out that $\vec{n}_1 \cdot \vec{n}_2 = 8$, but since we are doing our calculations in modulo 2, we see that $\vec{n}_1 \cdot \vec{n}_2 = 0$, and that $\vec{n}_1$ and $\vec{n}_2$ are an orthogonal basis for the Null (E).

Since $\vec{b}$ is winnable if and only if it belongs to the orthogonal complements of Null (E), then $\vec{b}$ is winnable if and only if it is orthogonal to the two vectors $\vec{n}_1$ and $\vec{n}_2$, thus proving Theorem 1.

B. Theorem 2

Theorem 2 states that if $\vec{b}$ is winnable, then the four winning strategies are

$$\begin{array}{rc} \vec{x} = & R\vec{b} \\ & R\vec{b} + n_1 \\ & R\vec{b} + n_2 \\ & R\vec{b} + n_1 + n_2 \end{array} \tag{10}$$

where $\vec{n}_1$ and $\vec{n}_2$ are the same as (6) and (7) and *R* is the product of the elementary matrices which perform the reducing row operation, RA=E .[1]

Since *E* has two independent variables, x_{24} and x_{25}, then the four configurations we can choose for them are

x_{24}	x_{25}
0	0
1	0
0	1
1	1

(11)

Let's choose $x_{24} = 0$ and $x_{25} = 0$. We can then see the product $E\,\vec{x} = \vec{x}$. We can then substitute in RA=E and $A\vec{x} = \vec{b}$ to get

$$\vec{x} = E\ \vec{x} = R\ A\ \vec{x} = R\ \vec{b}\ . \tag{13}$$

Therefore, one solution is $\vec{x} = R\,\vec{b}$. By looking at the other configurations for x_{24} and x_{25}, we can let x_{24} and x_{25} be as shown in (11). As a result, we can see that the four winning strategies are the four in (10).

C. Graphical User Interface

Using Theorems 1 and 2, we now have a way to mathematically check to see if a starting configuration is winnable. If it is winnable, we now have four equations that will find for a strategy that will turn all the lights out. To find the strategy with the fewest number of buttons pressed, you simply need to count the number of buttons that need to be pressed for each of the four strategies.

Since we are dealing with large matrices, 25 by 25, doing this computation by hand would be long and tedious. With the help of a computer algebra system or programming language capable of handling matrices like *Maple*, *Mathematica*, or *R*, one could compute the winning strategies. Due to my current knowledge, I did my computation in *R* and created a graphical user interface where one could enter a starting configuration and the GUI would return "No Solution" if there was no winning strategy or it would return the winning strategy with the fewest number of moves.

IV. Solving Using Light Chasing

A way that one could solve Lights Out without using a computer algebra system was first introduced to me by one of my colleagues [9]. Upon listening to him explain this method to me, which he called the "Binary Method," I did a little research and found that the method he was explaining was the Light Chasing Method.

The idea of the Light Chasing Method is to separate the 5 by 5 array into a 1 by 5 and a 4 by 5 array and then work on turning all the lights out in the 1 by 5 array. Then you would continue to separate the remaining array until you got down to a 1 by 5 array. An easy way to think about this is by turning out all the lights on the top row. One achieves this by simply pressing the buttons on the second row that are directly underneath a lit buttons on the top row. The top row will then have all its lights off. By repeating this step for the second, third, and fourth rows (i.e. chase the lights all the way down to the bottom row), you may have solved the puzzle, but it is more likely that there will now be some lights left on in the bottom row. If so, there are only seven possible configurations. Depending on which configuration you are left with, you will need to press some buttons in the top row. You can determine which buttons you need to press from the Figure 3.

The problem with the Light Chasing Method is that you will most likely end up pressing some buttons more that once, which means you won't solve the game in the fewest number of buttons pressed and will end up replaying the board. A simple solution to this is once you get down to your bottom row, restart the board and start by pressing the corresponding buttons in the top row to clear the board.

If you don't have the Figure 3 with you, my colleague that introduced me to this method explained that one should think of the five buttons in the top row as a five digit binary number staring at zero and increasing by one each time. If you press the buttons in the top row and it doesn't clear the board then continue on to the next binary number. For example, first try 00000, 10000, 01000, and 00100 where a one represents pressing the corresponding button. If you continue this, you will find that there are only 32 different ways to press the buttons in the top row. We proved earlier in this paper that there are 4 winnable solutions, so an eighth of the possible ways to press the buttons in the top row will clear the board. Using a geometric distribution and the "Binary Method", you have a 0.5512 probability of clearing the board within the first six attempts, with a mean of eight attempts.

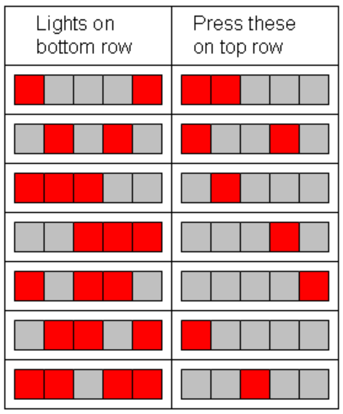

Figure 3. After chasing the lights down to the bottom row, you will have turned the lights out or have one of seven remaining configuration as shown in the left column. By pressing the buttons in the top row that are in the right column of the row of the remaining configuration and repeat the Chasing Lights Method you will turn all the lights out.

The other problem with solving lights out using Light Chasing is there is no way of checking to see if a board is winnable unless you go through all 32 different combinations.

V. Changes to the Game

So far this paper has just talked about the original 5 by 5 Lights Out, but since its release in 1995, Tiger Toys have created several other variations to the game.

The game Lights Out Cube consists of a 3 by 3 by 3 cube of lights and is played just like ordinary lights out. When you press a button at the edge of a 3x3 square, the affected lights wrap around onto the adjacent faces.

The Lights Out Keychain, also referred to as Lights Out Mini, is a 4 by 4 array of lights that plays on a torus. Playing on a 4 by 4 torus there is a unique board because every possible starting configuration is solvable and each solution is unique.

Lights Out Deluxe is a version of Lights Out played on a 6 by 6 array and offers some different game variations. These variations are slight changes in the rules. One variation is called Lit Only where you are only allowed to press buttons that are currently on.

Tiger Toys' latest Lights Out is called Lights Out 2000. This game plays 3-state Lights Out. The lights cycle through red, green, and off. This creates a significantly different and challenging game.

There are also hundreds of different variations that could be made to the game. First of all, the board could take on almost any shape it wants. Whether it is an *n* by *n* board, a circle, or a randomly scattered board, as long as you know which buttons toggle when you press a given button, you can play Lights Out.

VI. Conclusion

This paper has shown a variety of different methods that can be used to solve the electronic game Lights Out. One could use a computer algebra system capable of handling matrices where you know you can get an answer right away or use a guessing method like the "Binary Method" where you keep trying different sequences until onefinds a solution.

Having read this paper, I hope that I have inspired you to search the internet for an online version of Lights Out so you can practice the methods that I have talked about or a method of your own. If reading this paper has made you ambitious enough, go buy a microcontroller and try to recreate the game Lights Out.

Acknowledgment

I would like to acknowledge and thank the others members of my group, Jon Palecek and Prasoon Saurabh, for their input and help on my research. I would also like to thank our faculty advisor Professor Timothy Hess for his input and help with programming. I would also like to thank Ross Lewis for a great conversation about the game and introducing me to a different method for solving the game. Additionally, I would like to acknowledge Professor David Scott for helping me with various terms and calculations.

References

[1] Turning Lights Out with Linear Algebra, by Marlow Anderson and Todd Feil Mathematics Magazine, Vol. 71, No. 4 (Oct., 1998), pp. 300-303 Mathematical Association of America.
[2] An Easy Solution to Mini Lights Out, by Jennie Missigman and Richard Weida *Mathematics Magazine* © 2001 Mathematical Association of America.
[3] Two Reflected Analyses of Lights Out, by Óscar Martín-Sánchez and Cristóbal Pareja-Flores *Mathematics Magazine* © 2001 Mathematical Association of America.
[4] C. Haese, paper in the sci.nath newsgroup in 1998, available at http://www.math.niu.edu/-rusin/known- math/98/lights-out
[5] Hohn, Franz E. *Introduction to Linear Algebra*. New York: The Macmillian Company, 1972. 161-188. Print.
[6] K. Banr, Lights Out Fan Club, at http://gbs.mit.edu/-kbarr/lo/
[7] D.L. Stock, Merlin's magic square revisited, Amer Math. Monthly 96 (1989), 608-610.
[8] java version of the commercially marketed game *Lights Out* at http://www.whitman.edu/mathematics/lights_out/
[9] Barile, Margherita. "Lights Out Puzzle." From *MathWorld*--A Wolfram Web Resource, created by Eric W. Weisstein. http://mathworld.wolfram.com/LightsOutPuzzle.html
[10] Lewis, Ross. Conversation with about alterative solution

Matthew A. Madsen was born in Reedsburg, Wisconsin on November 7, 1988. He is currently pursuing his Baccalaureate degree in Mathematics and Physics from Ripon College in Ripon, Wisconsin and will be graduating in May 2011 after completing his final semester.

He is currently employed by the Mathematics and Computer Science Department and the Physics Department as a Departmental Assistant while working part-time at Webster's Pick 'N Save in Ripon, Wisconsin as customer service and terminal operator. He has spent several years working as a lifeguard at the Wilderness Resort in Wisconsin Dells.

Mr. Madsen is a member of the Society of Physics Students. In his free time he enjoys playing video games, hunting, watching movies and sports, and spending time with family and friends.