now enliven traditional topics. Tubes generated by specifying a curve and cross section occur naturally in a variety of settings, and their construction is an excellent illustration of some of the basic ideas in vector calculus. The graphical capabilities of systems like *Mathematica* and *Maple* enable us to see the results of these constructions, complementing traditional analytic and geometric techniques. On the other hand, while technology can enhance the standard curriculum, it is not a replacement. The parameterization of tubes as above is based on local orthonormal coordinate systems; so it is natural to exploit this geometry in the analysis. Indeed, computations are often impossible otherwise, even given the power of a computer algebra system.

REFERENCES

1. A. Gray, *Modern Differential Geometry of Curves and Surfaces*, CRC Press, Boca Raton, FL, 1993.
2. J. R. Munkres, *Analysis on Manifolds*, Addison-Wesley Publishing Company, The Advanced Book Program, Redwood City, CA, 1991.
3. J. Stewart, *Calculus, 3rd Edition*, Brooks/Cole Publ. Co., Pacific Grove, CA, 1994.

Two Reflected Analyses of Lights Out

ÓSCAR MARTÍN-SÁNCHEZ
CRISTÓBAL PAREJA-FLORES
Departamento de Sistemas Informáticos y Programación
Universidad Complutense de Madrid
Madrid, Spain

The device was described to us as a beeping hand-held electromechanical puzzle, with buttons that turned lights on and off. But we subsequently found that it was nothing of the sort. After playing several games, we felt the urge to open the screws and peer inside. What we found packed in this device really had nothing to do with mechanics or electronics at all ...

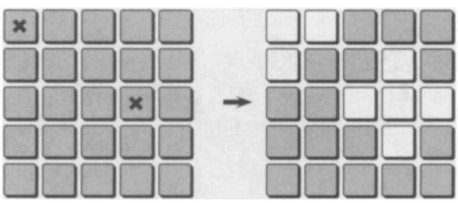
As someone with an *analytic mind* might point out, what is packed inside is strategy and empirical reasoning. To solve the puzzle, one needs nothing more than what is needed to solve a jigsaw puzzle: the human mind, some methodical work, and a little care.

As someone with *mathematical understanding* might point out, what is packed inside is a combination of matrices, vector spaces and scalar products. Indeed, this is just an instance of a well-known algebraic model—a system of linear equations—solvable with standard mathematical tools.

Two analyses follow: the left column addresses a person who is interested and methodical; the right one, a mathematician. These analyses are mirror images of one another: the concepts, examples, and figures on each side are designed to enrich their counterparts. Thus, for greater enjoyment and better understanding, we recommend a parallel reading of the two columns, passing fearlessly through the looking glass in the gap between the columns.

Our aim is to provide the reader with an understanding of the game, efficient algorithms to know when the game can be solved, and also how to find the solution. Our work is partially based on an article in the MAGAZINE by Anderson and Feil [**1**], but differs from that analysis in offering a way to solve the puzzle with the game in hand, without needing a computer or even pencil and paper. It also has elements of the theory of σ^+-automata [9, 3]. A collection of electronic resources for *Lights Out* can be found at http://www.maa.org/pubs/mathmag.html.

Rules of the game *Lights Out*® (Tiger Electronics) is a puzzle sold in toy stores. It consists of a 5×5 board of cells, where each cell is simultaneously a light and a button. Each light can either be on (white in the pictures) or off (gray). Pressing a button (marked with a cross in the figure), changes the on/off state of that light and also the lights of its horizontal and vertical neighbors. For example, if we start with all the lights off, then pressing the first button in the first row, and the fourth button in the third row, changes the lights according to the figure:



Such are the rules of the game.

The aim of the game is, starting with any given state of the board, to turn all the lights off (or *out*, if you prefer).

Although we are talking here about 5×5 boards, we do so in such a way that our methods can be applied to any $m \times n$ board.

Statement of the problem *Lights Out* is a problem in matrix algebra. We work with vectors of $(\mathbb{Z}_2)^{25}$. We are given the following boxed matrix R, which encodes the rules of the puzzle:

$$R = \begin{pmatrix} A & I & O & O & O \\ I & A & I & O & O \\ O & I & A & I & O \\ O & O & I & A & I \\ O & O & O & I & A \end{pmatrix}, A = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix},$$

where I and O are the 5×5 identity and zero matrices. For a vector $\vec{p}$, which we call a *press* vector, we calculate $R\vec{p}$, called the *effect* of $\vec{p}$. The effect of $\vec{p}$ is added (modulo 2) to vector $\vec{s}$, the initial *state*, to obtain a new state $R\vec{p} + \vec{s}$. For example, with the press vector that has 1s only in the 1^{st} and 14^{th} components, the null state is changed as follows:

$$R \times \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} + \vec{0} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

(When needed, 25-component vectors are arranged in a 5×5 fashion.) The aim is, given a state $\vec{s}$, to find $\vec{p}$ such that $R\vec{p} + \vec{s} = \vec{0}$. Although we are talking here about 25-component vectors, our methods can be applied to vectors of any $m \times n$ size, and matrices resembling R.

First remarks To solve the puzzle we must press a number of buttons; during this process, some lights are switched several times, but we are only interested in the final result.

Pressing a button twice has no effect. Also, solving a given state is the same as reaching it from a fully unlit board: the same presses are needed.

Pressing one button and then another one has the same effect as pressing them in reverse order. This is because the final state of a cell depends only on (the initial state and) the number of buttons pressed that switch it; it has nothing to do with the order.

First remarks First, note that the names we give to vectors and other elements (like press vectors, effects, states) are just a way of speaking: no physical meaning is needed.

An equivalent statement of the problem is, given $\vec{s}$, to find $\vec{p}$ such that $R\vec{p} = \vec{s}$. This is a well-known algebraic problem. (Observe that, modulo 2, $\vec{s} = -\vec{s}$.)

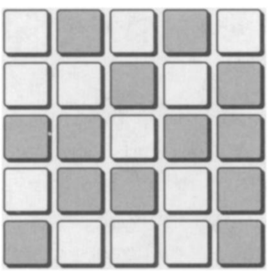
Observe that it makes sense to solve the problem, for a given $\vec{s}$, in more than one step: starting from the null state, $\vec{0}$, we might want to press $\vec{p}_1$, so that the new state, $\vec{s}_1$, is of a more convenient form, then press $\vec{p}_2$, etc.:

$$\vec{0} \xrightarrow{\vec{p}_1} \vec{s}_1 \xrightarrow{\vec{p}_2} \cdots \longrightarrow \vec{s}$$

Therefore, a set of presses has the same effect if we remove all pairs of equal presses. Such a set of presses, where no cell is pressed more than once, we call a *procedure*.

The puzzle can be solved by trying out all the possible combinations. Since, for every button, we have to choose whether to press it or not, we have a huge total of 2^{25} procedures. We seek to simplify the solving method for someone with the game in hand.

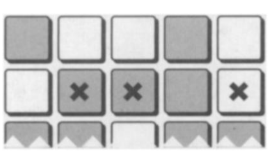
Reducing the search Given a state that we want to solve,



suppose we arbitrarily choose some buttons in row 1 (the top one), as below:

After pressing them, some lights in row 1 may be on

and the only way to turn them off (without further presses in row 1, and without switching on the lights that are off) is to press the buttons in row 2 that are exactly under any lights that are on in row 1:



The state of row 2 then determines what must be pressed in row 3, and so on through the rows. Therefore, given a particular state and chosing a set of buttons to press in the first row, the rest of the procedure is determined:

The solution would be the sum of the $\vec{p}_i$s:

$$\vec{0} \xrightarrow{\vec{p}_1+\vec{p}_2+\cdots} \vec{s}$$

As this is a known algebraic problem, we could use standard methods to solve it, as used in [**1**]. But these standard methods use matrices in general, whereas here we seek to exploit the particular features of R.

Finding linear dependences The problem can be written as follows:

$$R\cdot\begin{pmatrix}\vec{p}_{1\bullet}\\ \vec{p}_{2\bullet}\\ \vec{p}_{3\bullet}\\ \vec{p}_{4\bullet}\\ \vec{p}_{5\bullet}\end{pmatrix}=\begin{pmatrix}\vec{s}_{1\bullet}\\ \vec{s}_{2\bullet}\\ \vec{s}_{3\bullet}\\ \vec{s}_{4\bullet}\\ \vec{s}_{5\bullet}\end{pmatrix}$$

where, for the sake of clarity, $\vec{p}$ and $\vec{s}$ have been divided into subvectors $\vec{p}_{i\bullet}$ and $\vec{s}_{i\bullet}$, each consisting of 5 components.

Manipulating the previous equation leads to something more convenient: an expression for $\vec{p}_{1\bullet}$ depending only on $\vec{s}$, and others for $\vec{p}_{2\bullet}\ldots\vec{p}_{5\bullet}$ depending only on $\vec{p}_{1\bullet}$ and $\vec{s}$.

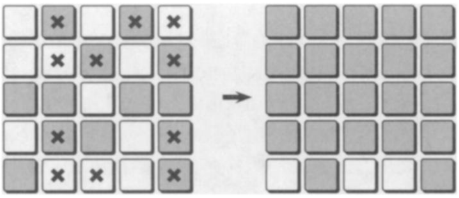
The equation $R\vec{p}=\vec{s}$ is equivalent to $J\vec{p}=(R+J)\vec{p}+\vec{s}$ for any matrix J. Using, in particular,

$$J=\begin{pmatrix}O & I & O & O & O\\ O & O & I & O & O\\ O & O & O & I & O\\ O & O & O & O & I\\ O & O & O & O & O\end{pmatrix}$$

we find:

$$\begin{pmatrix}\vec{p}_{2\bullet}\\ \vec{p}_{3\bullet}\\ \vec{p}_{4\bullet}\\ \vec{p}_{5\bullet}\\ \vec{0}\end{pmatrix}=\begin{pmatrix}A & O & O & O & O\\ I & A & O & O & O\\ O & I & A & O & O\\ O & O & I & A & O\\ O & O & O & I & A\end{pmatrix}\begin{pmatrix}\vec{p}_{1\bullet}\\ \vec{p}_{2\bullet}\\ \vec{p}_{3\bullet}\\ \vec{p}_{4\bullet}\\ \vec{p}_{5\bullet}\end{pmatrix}+\begin{pmatrix}\vec{s}_{1\bullet}\\ \vec{s}_{2\bullet}\\ \vec{s}_{3\bullet}\\ \vec{s}_{4\bullet}\\ \vec{s}_{5\bullet}\end{pmatrix}$$

Here, each $\vec{p}_{i\bullet}$ depends only on the subvectors of $\vec{p}$ and $\vec{s}$ with indices smaller than its own. Thus, we can use the equation from the first row to remove $\vec{p}_{2\bullet}$ from the second, then use the second row to remove $\vec{p}_{3\bullet}$ from the third, and so on.



But note that this *gathering* procedure only guarantees to turn off all the lights in the four upper rows: lights in the fifth row may be still on after this procedure.

We can thus deal with the solution to a given board in three steps:

- first, we apply the gathering procedure down to the last row, without doing any presses in row 1;
- then, we look for presses in the first row that, *when gathered*, would turn the remaining lights off;
- finally, we press those buttons in the first row, and gather the result.

But it could be that no set of presses in the first row can actually fulfill this task. In this case, there is no solution. We will return to this issue later.

The question is: which buttons must be pressed in the first row? In order to find out, for each button in the first row we start with an unlit 5×5 board, press that button, and then gather the resulting position down. These are the results in the *fifth* row, for each button in the *first* one:

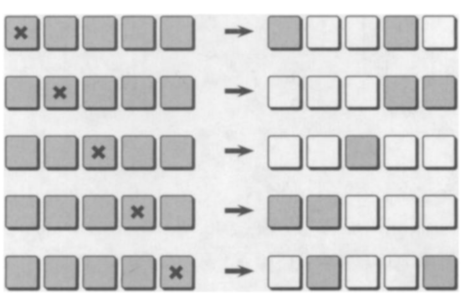


Figure 1

Therefore, $\vec{p}_{2\bullet} \dots \vec{p}_{5\bullet}$ are expressed depending only on $\vec{s}$ and $\vec{p}_{1\bullet}$:

$$\begin{pmatrix} \vec{p}_{2\bullet} \\ \vec{p}_{3\bullet} \\ \vec{p}_{4\bullet} \\ \vec{p}_{5\bullet} \\ \vec{0} \end{pmatrix} = \begin{pmatrix} B_1 & B_0 & O & O & O & O \\ B_2 & B_1 & B_0 & O & O & O \\ B_3 & B_2 & B_1 & B_0 & O & O \\ B_4 & B_3 & B_2 & B_1 & B_0 & O \\ B_5 & B_4 & B_3 & B_2 & B_1 & B_0 \end{pmatrix} \begin{pmatrix} \vec{p}_{1\bullet} \\ \vec{s}_{1\bullet} \\ \vec{s}_{2\bullet} \\ \vec{s}_{3\bullet} \\ \vec{s}_{4\bullet} \\ \vec{s}_{5\bullet} \end{pmatrix}$$

where $B_0 = I$, $B_1 = A$ and $B_{n+2} = A \times B_{n+1} + B_n$. Now, the equation

$$B_5\vec{p}_{1\bullet} = B_4\vec{s}_{1\bullet} + B_3\vec{s}_{2\bullet} + B_2\vec{s}_{3\bullet} + B_1\vec{s}_{4\bullet} + B_0\vec{s}_{5\bullet}$$

taken from the bottom row of the former matrix, involves only the $\vec{p}_{1\bullet}$ part of $\vec{p}$.

We can thus deal with this equation in three steps:

- first, we calculate the right-hand side, which only depends on $\vec{s}$; we will call this vector *gather*$(\vec{s})$ for brevity;
- then, we try to find $\vec{p}_{1\bullet}$ such that $B_5\vec{p}_{1\bullet}$ equals the now-known *gather*$(\vec{s})$;
- finally, we calculate $\vec{p}_{2\bullet} \dots \vec{p}_{5\bullet}$ directly from $\vec{p}_{1\bullet}$ and $\vec{s}$, using the equations in the four upper rows of the previous matrix.

There exists some $\vec{s}$ for which no $\vec{p}_{1\bullet}$ satisfies the equation above. In such cases, the problem has no solution. We will study this issue later.

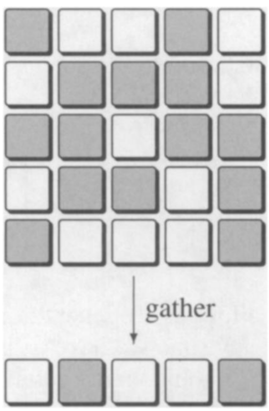
The only remaining question is how to obtain $\vec{p}_{1\bullet}$ such that it satisfies the equation

$$B_5\, \vec{p}_{1\bullet} = gather(\vec{s}).$$

The matrix B_5, that will be necessary below, is the following:

$$B_5 = A^5 + A = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

We have thus shifted from looking for which of 25 buttons need to be pressed in order to turn 25 possible lights off, to the simpler case of 5 buttons and 5 lights; this simplifies the search for solutions. In the following example, the gathering happens to produce the effect we calculated for button 5 above; thus, the solution is achieved by pressing button 5 in row 1 and gathering.



In a sense, we are now playing on a 1×5 board, with very different rules for how the lights change when a button is pressed, as can be seen in FIGURE 1. The new game inherits many properties from the original one: the order of presses is not relevant, and double presses have no effect.

Every solution to this game (1×5) gives a solution to the original game (5×5), and *vice versa*. Non-solvable states (if any) also correspond; the same can be said about states with more than one solution (if any). Procedures in the large board are called the *expansion* of the ones in the small one with respect to the state being solved.

The linear dependences found allow us to see the 25-dimensional problem

$$R\,\vec{p} = \vec{s},$$

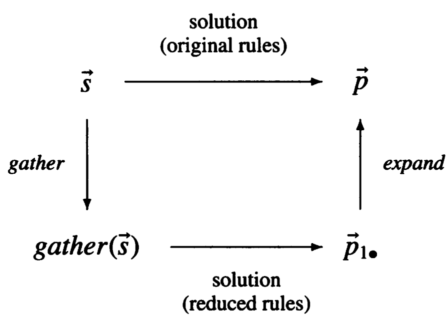
as the 5-dimensional one

$$B_5\,\vec{p}_{1\bullet} = gather(\vec{s}).$$

As an example, the following state is reduced as shown below, leading to a much smaller problem:

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix} \quad \Big\downarrow \; gather \quad \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Each solution to this problem gives one and only one solution to the original problem, and *vice versa*. Thus, the following diagram is commutative:



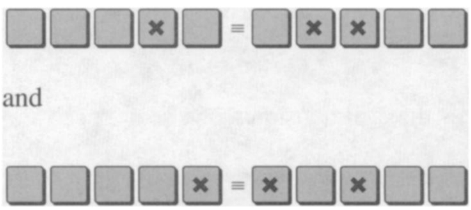
We call the 25-component vectors the *expansion* of the 5-component vectors with respect to the state being solved: $\vec{p} = expand(\vec{p}_{1\bullet}, \vec{s})$.

In what follows, we are mainly playing with reduced boards, the results and conclusions of which correspond to those of the large boards.

From the equivalence of these problems, we get that $null(R)$ and $null(B_5)$ have the same dimension. In the next section, we explore this issue further.

Neutral but useful procedures Looking at the rules of the reduced game in FIGURE 1, some equivalences between procedures can be found, for instance:

Image and null space of R We can perform Gauss-Jordan reduction on B_5 to get $XB_5 = E$, obtaining the matrices:



Several useful conclusions can be drawn from here. To begin with, these equivalences show that it is never necessary to push the two buttons farthest to the right: a solution using those buttons could be replaced by an equivalent one in which they are not used at all. For the purposes of our solution, we could pretend they were broken.

Also, we note that if two equivalent procedures are performed in turn, one undoes the other, and the overall result is null. The composition of these two procedures will be called a neutral procedure.

Combining each pair of equivalent procedures above gives two different neutral procedures. A third is obtained by combining these two. There is also a fourth, the trivial one, in which no buttons are pressed. A systematic search proves that there are no others.

When any procedure is followed by (or composed with) a neutral one, an equivalent procedure results. Therefore, when there is a solution for a given state, there are actually four solutions (as there are four neutral procedures).

$$X = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

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

We note that $null(E)$ equals $null(B_5)$, and its dimension is 2, and therefore so is $null(R)$.

Computation produces a basis of $null(E)$, the vectors:

$$\vec{i} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} \quad \text{and} \quad \vec{j} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}$$

These allow us to generate additional solutions any time we have a single one: If $\vec{p}$ is a (5-component) solution for $\vec{s}$, there are three other solutions:

$$\vec{p}+\vec{i} \qquad \vec{p}+\vec{j} \qquad \vec{p}+\vec{i}+\vec{j}$$

Inspecting the form of $\vec{i}$ and $\vec{j}$, we see that one of these four solutions will have zeroes in the fourth and fifth components.

Is it even possible? We can now find out if a state is, or is not, solvable with just 8 tries. But we can achieve a more direct solution using more subtle reasoning.

First, note that the games, both the reduced and the original, have a certain symmetry: if button x toggles light y, then button y toggles light x. Any neutral procedure, because it is neutral, presses an even number of buttons in the neighborhood of *any* cell, leaving it unchanged. But then, symmetrically, pressing any button switches an even number of cells in N, the set of cells pressed in a neutral procedure. Therefore, the parity

Does a solution exist? We can reduce our search for solutions to 3-component vectors. Therefore, only 2^3 states are reachable out of the total posible 2^5 states.

In other words, the image of our matrix has the dimension of 3; we know this because the order of a square matrix (5 in our case) equals the dimension of its null space (2) plus the dimension of its image.

Therefore, not every state can be solved; it is thus worthwhile to look for criteria to identify the solvable states.

Algebra tells us that, given a symmetric matrix (like B_5), its image is orthogonal to its null space. So, for a given $\vec{s}$, there of the lit cells in N cannot be changed by any press.

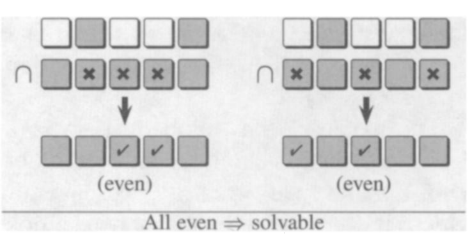
So, from the fully unlit state, we can only get states with an even number of cells in common with any such neutral procedure N. Therefore, a state is solvable only if it has an *even* intersection with any neutral procedure.

For a given neutral procedure, this test halves the number of possibly solvable states. Since any neutral procedure is the composition of the other two, it is enough to test it for two of them. Thus, we can halve the number of solvable states twice, and mark three quarters of the states as unsolvable.

On the other hand, each solvable state can be solved by means of exactly four different procedures. Therefore, one quarter of the states are solvable: those that pass the previous test.

Let's choose these two neutral procedures to perform the parity test:

We can use these results to show that the example we used before is solvable. The target state had lights 1, 3, and 4 on; comparing this with the first neutral state above, we find two lights in common: the third and fourth; comparing with the second again gives an even number: the first and third buttons. Thus the state is solvable:



How to solve it? Now we have a puzzle with a state we want to solve. How are we going to do it? We know we need only focus on the three buttons on the left. Their effects are as follows:

exists a solution $\vec{p}$ for the equation

$$B_5\vec{p} = \vec{s},$$

if and only if $\vec{s}$ is orthogonal to the null space of the matrix. Because of the equivalence between the expanded and reduced problems, we know that this condition can be checked either with 5×5 matrices, or 25×25 ones.

Working with the convenient smaller system, for $\vec{s}$ to be solvable, the scalar product $\vec{s} \cdot \vec{n}$ should be 0 for every $\vec{n} \in null(B_5)$. And it is enough to test it for $\vec{n}$ in the basis of $null(B_5)$. This amounts to checking the following equations:

$$s_2 + s_3 + s_4 = 0$$

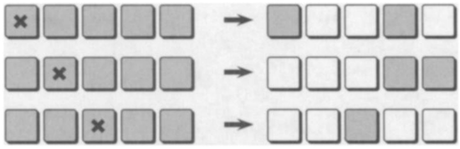
$$s_1 + s_3 + s_5 = 0.$$

Let's use this theory to show that the example we used before is solvable. In the figures below, in each box, we calculate the scalar product of the state under consideration and one of the neutral procedures. We find that both are null, that is, the two equations above are satisfied. Therefore, the state is solvable. And so is the corresponding expanded state whose *gather* yields this reduced state.

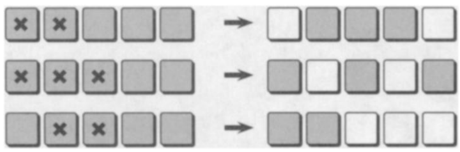
$(1, 0, 1, 1, 0)$ $\cdot\ (0, 1, 1, 1, 0)$ $\parallel$ $0+0+1+1+0$	$(1, 0, 1, 1, 0)$ $\cdot\ (1, 0, 1, 0, 1)$ $\parallel$ $1+0+1+0+0$
$(s_2 + s_3 + s_4 = 0)$	$(s_1 + s_3 + s_5 = 0)$

All satisfied $\Rightarrow$ solvable

Solution algorithm Consider again the matrix



Among the 8 possible ways of combining the presses of these three buttons, three combinations stand out:



Each is useful because it only switches one of the three lights to the left. This provides the first way to find a candidate for a solution, without using buttons 4 and 5. This candidate only depends on the state of the three first lights. We call it a candidate rather than a solution, because we have no assurance that the last buttons will turn out the right way. We will consider this below.

For a different way of finding the solution, look again at the effects of the presses of the first three buttons at the beginning of this section. Note that only button 1 can switch exactly one of the first two lights. This tells us that button 1 must be pressed if and only if exactly one of the first two lights is on. For button 2, we have to count how many of the first three lights are on: we press button 2 if and only if this is odd. Likewise, pressing button 3 depends on the parity of the set containing the second and third lights. This approach is also used in [**8**].

This gives us a second way to solve any puzzle, which we state for the nonreduced game: gather the state down to row 5; then, in row 1, press whichever buttons are required by the parity rules given above, based on which lights remain on in the last row; gather the new state down again. If the initial state was solvable, we have solved it in this way. A way to remember this technique is to note that the lights governing the press of a button are the ones in its *mini-neighborhood*, that is, the neighbors in the subset of the three cells on the left.

$$B_5 = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

of the reduced problem.

Following the approach of [**1**] we will look for a solution to our reduced problem such that $p_4 = p_5 = 0$.

Using the Gauss-Jordan reduction of B_5, we have

$$\vec{p} = E\vec{p} = XB_5\vec{p} = X\vec{s}.$$

This allows us to compute one of the solutions easily; the other three are obtained by adding the vectors in the null space of the matrix to $\vec{p}$.

Remembering the matrix

$$X = \left(\begin{array}{ccc|cc} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ \hline 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{array}\right),$$

we can express the form of the solutions as follows:

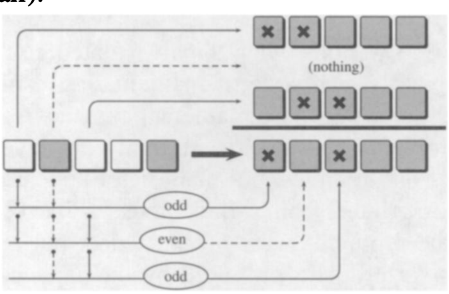
$$\begin{aligned} p_1 &= s_1 + s_2 \\ p_2 &= s_1 + s_2 + s_3 \\ p_3 &= \phantom{s_1 +{}} s_2 + s_3 \end{aligned}$$

In summary, to calculate a solution to any 25-component state $\vec{s}$ in the nonreduced problem, we calculate the first three components of $gather(\vec{s})$; then find p_1, p_2, and p_3; then, finally, the solution $expand(\vec{p}_{1\bullet}, \vec{s})$.

The matrix X has, embedded in it, a lot of information about the reduced problem. For the sake of clarity, we have divided it into three boxes above. Look first at the lower box. It determines, for a given state $\vec{s}$, the two last components of the presses $\vec{p}$ that solve it. But we have assumed above that $p_4 = p_5 = 0$. This gives another view of the condition for $\vec{s}$ to be solvable, because the two rows in this box are a basis of $null(B_5)$.

The relationship between these two solutions above is interesting: the set of cells that governs whether to press a button v constitutes the candidate solution for cell v. To prove this, let's reason about the symmetry of the game, and the parities of certain sets of cells. Suppose that v is the only button that changes the parity of the number of lights on in a set U. This means that v is the only cell whose neighborhood has an odd number of cells in U. And, therefore, U is *the* candidate solution for v.

This result can be applied to boards of any size. If we work out in advance the solutions (3, in our case) to the appropriate cells, possibly by trial and error, this is enough to solve, with a few further calculations, any other state of the board. The following figure shows the two ways previously described to find a solution: considering lights one by one (upper half), or considering buttons one by one (lower half):



Up to this point, we have not considered lights 4 and 5. We know that the first three lights uniquely determine a candidate solution (that is, something to try). When a solution does exist, all the lights are going to be left off, and our candidate will be an actual solution. If no solution exists, and thus some neutral procedure fails the parity criterion, then either one or both of the last two lights will remain on. The light that remains on denotes which of the neutral procedures fails the parity criterion: light 4 corresponds to the three central buttons, and light 5 to the three alternate ones.

Now, consider the upper right box of X. It is null, which means that the solution we are going to calculate does not depend on the last two components of $\vec{s}$. Thus, the last two components of a state are only useful to check whether the state is solvable or not.

So, in a sense, we have further reduced our original problem to one of dimension 3, whose solution is given by the upper left box of X. The rules of this new problem are given by the 3×3 box in the upper left corner of B_5. This new problem always has a solution; once we know it, we can attempt to solve the others. For our example:

$$\begin{aligned} p_1 &= s_1 + s_2 &&= 1 + 0 &&= 1 \\ p_2 &= s_1 + s_2 + s_3 &&= 1 + 0 + 1 &&= 0 \\ p_3 &= \quad\; s_2 + s_3 &&= \quad\; 0 + 1 &&= 1 \end{aligned}$$

But it is only an attempt. We calculate the solution just by looking at the first three components of $gather(\vec{s})$, so the other two can result 0 or 1. They are guaranteed to be null if there is a solution, that is, if the state is orthogonal to the null space of the matrix. When there is no solution, as can be deduced from X, the vector of the basis of the null space for which the orthogonality fails is denoted by the component, fourth or fifth, that remains nonnull.

Acknowledgment. We thank Gustavo Luengo, Pedro Palao, and Cristina Rodríguez for their helpful comments on a previous draft of this article; we are also indebted to the LaTeXmaster Pedro PALAO for his nice collection of macros.

Benjamin Lotto provided us with information about this game, as did Robert A. Moeser and Anya Hillery, who also helped to improve the English of the paper.

The authors of the left column also express their gratitude to these people for their contribution.

REFERENCES

1. M. Anderson and T. Feil, Turning lights out with linear algebra, this MAGAZINE **71** (1998), 300–303.
2. K. Barr, *Lights Out Fan Club*, at http://gbs.mit.edu/~kbarr/lo/
3. R. Barua and S. Ramakrishnan, σ-game, σ^+-game and two-dimensional additive cellular automata, *Theoretical Computer Science* **154**:2 (1996), 349–366.
4. C. Haese, paper in the sci.math newsgroup in 1998, available at http://www.math.niu.edu/~rusin/known-math/98/lights_out
5. B. Lotto, A computer puzzle as a paradigm for doing mathematics, *Vassar Quarterly*, Winter 1996, 18–23.
6. B. Lotto, personal letter, May 1999.
7. Ó. Martín-Sánchez and C. Pareja-Flores, *A Visual Basic Implementation of Lights Out*, available at http://dalila.sip.ucm.es/miembros/cpareja/lo
8. D.L. Stock, Merlin's magic square revisited, *Amer. Math. Monthly* **96** (1989), 608–610.
9. K. Sutner, Linear cellular automata and the garden-of-Eden, *The Mathematical Intelligencer* **11**:2 (1989), 49–53.

An Equilateral Triangle with Sides through the Vertices of an Isosceles Triangle

FUKUZO SUZUKI
Gunma College of Technology
580 Toriba Machi Maebashi
Gunma Japan 371-8530
suzuki@nat. gunma-ct.ac.jp

In the years 1603–1867, known as the Edo period, when Japan isolated itself from the western world, the country developed its own style of mathematics, especially geometry. Results and theorems of traditional Japanese mathematics, known as *Wasan*, were usually stated in the form of problems; these were originally displayed on wooden tablets (*Sangaku*) hung in shrines and temples, but many later appeared in books, either handwritten with a brush or printed from wood blocks. (See [**2**], [**3**], and [**4**] for more details.) Solutions to the problems were not provided, but answers were sometimes given. One of these is a problem proposed by the Japanese mathematician Tumugu Sakuma (1819–1896).

In this note, we take up a generalization of his problem: In FIGURE 1, triangle $\triangle ABC$ is an equilateral triangle and each of the sides CA, AB, BC (or their extensions) passes through three vertices L, M, N of an isosceles triangle $\triangle LMN$ with $ML = MN$. Find a relation among LA, MB, and NC. Our solution reveals an invariant property of this configuration.

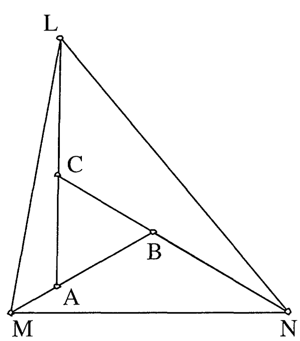


Figure 1