

# Lights Out Animations

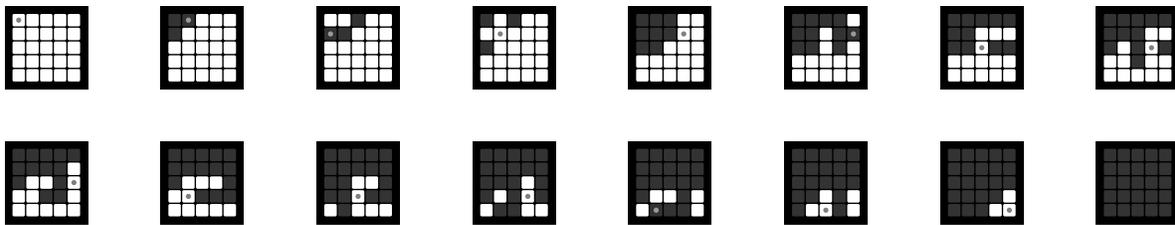
Robert Bosch  
Oberlin College  
rbosch@oberlin.edu

## Abstract

Given two grayscale target images, we produce two configurations of a Lights Out game board, each accessible from the all-on configuration and each closely resembling its target image. We then find a collection of cells that when clicked (in any order) will produce an animation that transforms the first configuration into the second.

## Introduction

In a previous life, before I began my investigations into how mathematical optimization techniques can be used to design visual artwork, I wrote the *Mind Sharpener* puzzle column for *Optima*, the newsletter of what was then called the Mathematical Programming Society and is now called the Mathematical Optimization Society. Inspired by the work of Martin Chlond [2], I devoted one installment of my column to integer programming (IP) models for solving variants of the Lights Out puzzle [1]. Lights Out is a handheld electronic solitaire game that was manufactured by Tiger Electronics. The player's task is to turn off all 25 lights that make up a  $5 \times 5$  grid of cells. Clicking on a cell activates a toggle switch that causes the state of the cell to change from on to off or from off to on. The complication is that clicking a cell also activates toggle switches for the cell's orthogonal neighbors—the cells immediately above it, below it, to its left, and to its right. To turn off all the lights, the player must determine which cells to click. Clicking a cell more than once isn't necessary, and the order in which clicked cells are clicked doesn't matter. The challenge is to turn the lights off with as few clicks as possible. An optimal solution to the  $5 \times 5$  Lights Out puzzle is shown in Figure 1.



**Figure 1:** An optimal solution to the  $5 \times 5$  Lights Out puzzle.

In the present paper, we describe how IP models can be used to design Lights Out animations. We begin by reviewing the basic Lights Out IP model and showing how it can be used to (1) determine whether a black-and-white target image  $B$ , when interpreted as a Lights Out configuration, is accessible from the all-on configuration, and (2) if it is accessible, determine which cells should be clicked in order to produce the image with the least number of clicks. We then show how to modify the IP in order to produce an accessible configuration that most closely resembles an nonaccessible black-and-white image  $B$ . We conclude by describing how to modify the IP in order to design a Lights Out animation that resembles one black-and-white target image in its first frame and a second black-and-white target image on its final frame.

## The Image

We assume that our target image is an  $m \times n$  black-and-white image  $B$ . We let  $b_{i,j} \in \{0, 1\}$  denote the row- $i$ -column- $j$  entry of  $B$ , the brightness value of pixel  $(i, j)$ . If  $b_{i,j} = 0$ , then pixel  $(i, j)$  is black. If  $b_{i,j} = 1$ , then pixel  $(i, j)$  is white.

If we want to work with a grayscale target image, we can use downsampling and Floyd-Steinberg dithering to convert into a black-and-white target image.

## The Basic IP Model

Here we consider two related problems. First, we want to be able to determine if  $B$  is accessible from the all-on configuration. Second, if it is accessible, we want to determine how to produce it with the least number of clicks.

If we click cell  $(i, j)$ , cell  $(i, j)$  changes state and so does its orthogonal neighbors. We let  $C(i, j)$  stand for the set of cells that change state when we click cell  $(i, j)$ . Note that if cell  $(i, j)$  is an interior cell, then

$$C(i, j) = \{(i-1, j), (i, j-1), (i, j), (i, j+1), (i+1, j)\},$$

so  $|C(i, j)| = 5$ . If cell  $(i, j)$  is an edge cell, then  $|C(i, j)| = 4$ . If cell  $(i, j)$  is a corner cell, then  $|C(i, j)| = 3$ .

But which cells should we click? The basic IP model has a variable  $x_{i,j}$  for each cell  $(i, j)$ . The variable  $x_{i,j}$  equals 1 if we click on cell  $(i, j)$  and 0 if we do not. To reduce the “notational clutter” in our model, we define  $x(C(i, j))$  to be the sum of all variables  $x_{i',j'}$  for which  $(i', j') \in C(i, j)$ . Note that if cell  $(i, j)$  is an interior cell, then

$$x(C(i, j)) = x_{i-1,j} + x_{i,j-1} + x_{i,j} + x_{i,j+1} + x_{i+1,j}.$$

We refer to  $x(C(i, j))$  as the *change count* for cell  $(i, j)$ , as it counts the number of times that cell  $(i, j)$  changes state.

Because we are starting from the all-on (all white) configuration, we want cell  $(i, j)$  to change state an even number of times when  $b_{i,j} = 1$  (when pixel  $(i, j)$  of  $B$  is white) and an odd number of times when  $b_{i,j} = 0$  (when pixel  $(i, j)$  of  $B$  is black). This means that we want to find an assignment of 0s and 1s to the the  $x_{i,j}$  variables that satisfies

$$x(C(i, j)) \equiv 1 - b_{i,j} \pmod{2},$$

or equivalently,

$$x(C(i, j)) = 2y_{i,j} + 1 - b_{i,j} \quad \text{for some } y_{i,j} \in \{0, 1, 2\}.$$

Consequently, to determine if  $B$  is accessible from the all-on configuration—and to find the least number of clicks required to produce it if it is accessible—we can solve the following integer program:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^m \sum_{j=1}^n x_{i,j} \\ \text{subject to} & x(C(i, j)) = 2y_{i,j} + 1 - b_{i,j} \quad \text{for all } (i, j) \\ & x_{i,j} \in \{0, 1\} \quad \text{for all } (i, j) \\ & y_{i,j} \in \{0, 1, 2\} \quad \text{for all } (i, j). \end{array}$$

## An IP Model for Inaccessible Images

What if  $B$  is *not* accessible from the all-on configuration? If  $B$  is inaccessible, the basic IP model will be infeasible. If this is the case, we would like to determine which cells to click in order to produce an accessible configuration that most closely resembles  $B$ .

One way to do this is to introduce additional variables  $z_{i,j}$  into the basic IP model:

$$\begin{array}{ll}
 \text{minimize} & \sum_{i=1}^m \sum_{j=1}^n z_{i,j} \\
 \text{subject to} & x(C(i, j)) = 2y_{i,j} + z_{i,j} \quad \text{for all } (i, j) \text{ such that } b_{i,j} = 1 \\
 & x(C(i, j)) = 2y_{i,j} + 1 - z_{i,j} \quad \text{for all } (i, j) \text{ such that } b_{i,j} = 0 \\
 & x_{i,j} \in \{0, 1\} \quad \text{for all } (i, j) \\
 & y_{i,j} \in \{0, 1, 2\} \quad \text{for all } (i, j) \\
 & z_{i,j} \in \{0, 1\} \quad \text{for all } (i, j).
 \end{array}$$

Note that we have split the change count constraints into two groups. The first group is for the white pixels, those with  $b_{i,j} = 1$ . Because we are starting from the all-on configuration, we want the change counts for the white pixels to be even. If this is possible, each white pixel's  $z_{i,j}$  will equal 0. If it isn't possible to make a white pixel  $(i, j)$ 's change count even, its  $z_{i,j}$  will equal 1.

The second group is for the black pixels, those with  $b_{i,j} = 0$ , and similar reasoning applies. Because we are starting from the all-on configuration, we want the change counts for the black pixels to be odd. If this is possible, each black pixel's  $z_{i,j}$  will equal 0. If it isn't possible to make a black pixel  $(i, j)$ 's change count odd, its  $z_{i,j}$  will be 1.

Accordingly, we can think of the  $z_{i,j}$  variables as flags that identify errors. A  $z_{i,j}$  variable equals 1 when its pixel is the wrong color. By minimizing the sum of the  $z_{i,j}$  variables, we are minimizing the number of errors.

## Computational Experience

We have not performed extensive tests of the models. At the moment, it appears that basic IP model works well (i.e., solves quickly) on small-to-moderate instances (up  $20 \times 20$ ) associated with accessible images. If  $B$  is larger and is accessible from the all-on configuration, the basic IP model can take a disappointingly long time to detect accessibility, and it can take much longer to obtain an optimal set of cells to be clicked. If  $B$  is inaccessible, the basic IP model can take a substantial amount of time to conclude that it is inaccessible.

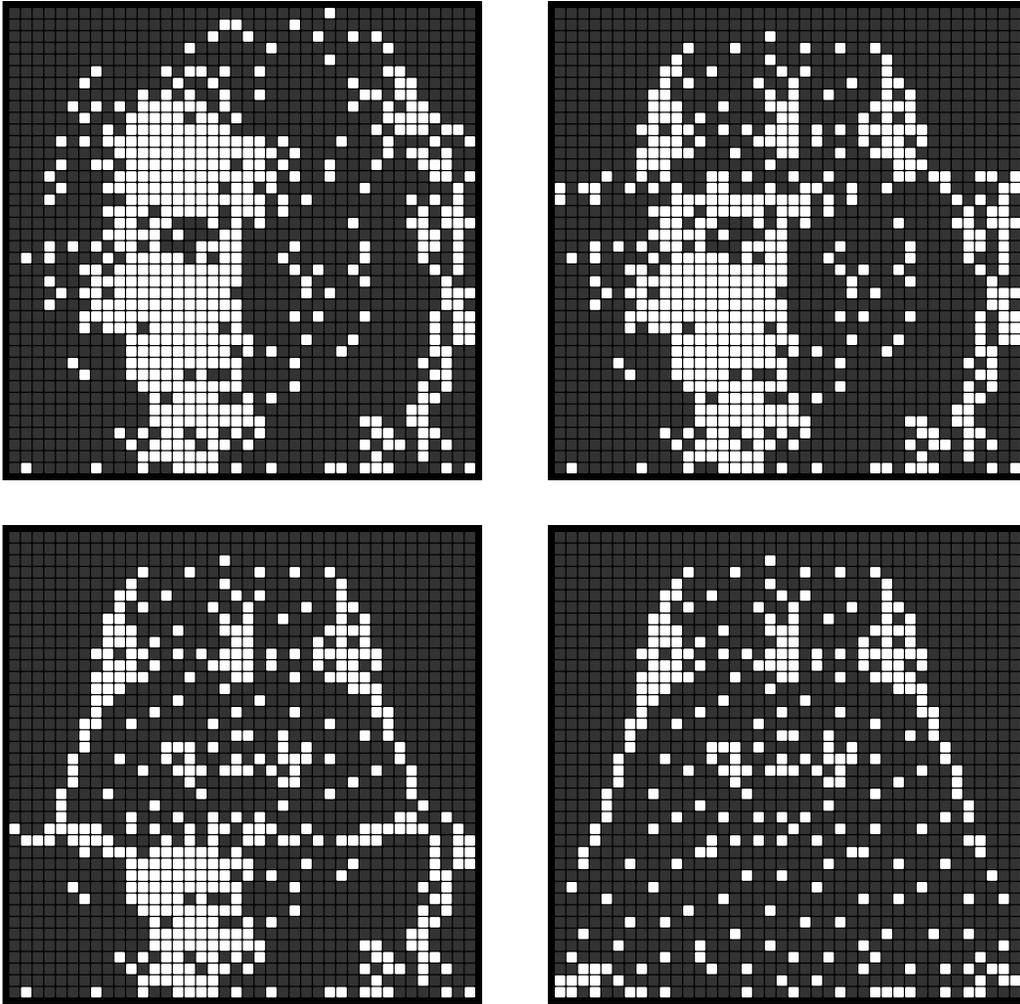
The modified IP produces low-quality (high-error) solutions quickly and often takes a considerable amount of time to produce high-quality (low-error) solutions. Fortunately, it is easy to design heuristics that produce high-quality solutions. The idea is to start with the black-and-white image and try to reach the all-on configuration. It is easy to see that all the errors can be pushed to the bottom row (or to the top row, or to the right column, or to the left column). The low-error solution can be used as an initial solution for the IP solver. Usually the IP solver will be able to improve upon the initial solution (often cutting the error in half) within a minute or two of computation.

## Animations

We now describe how to design a Lights Out animation that resembles one black-and-white target image in its first frame and a second black-and-white target image on its final frame. We begin by employing the

modified IP twice. The first time, we use it to find an accessible configuration that closely resembles the first target image. The second time, we use it to find an accessible configuration that closely resembles the second target image. Because both configurables are accessible from the all-on configuration, they are accessible from each other.

To design the animation, we apply the basic IP model to the configuration formed by taking the exclusive-or of the two accessible configurations. We form an initial solution by taking the exclusive-or of the two solutions's  $x_{i,j}$  values. An example animation is shown in Figure 2. The top left shows the first frame. The top right and bottom left are the 200th and 400th frames, respectively. The bottom left is the final frame.



**Figure 2:** “From Light to Dark” (four frames of a Lights Out animation). ©Robert Bosch, 2016.

## References

1. Bosch, R.A., 2000. Lights Out. *Optima (Newsletter of the Mathematical Programming Society)*, 64, p. 15. Available online at [mathopt.org/Optima-Issues/optima64.pdf](http://mathopt.org/Optima-Issues/optima64.pdf).
2. Martin J. Chlond and Toase, C.M., 2002. IP modeling of chessboard placements and related puzzles. *INFORMS Transactions on Education* 2(2), pp. 1-11.