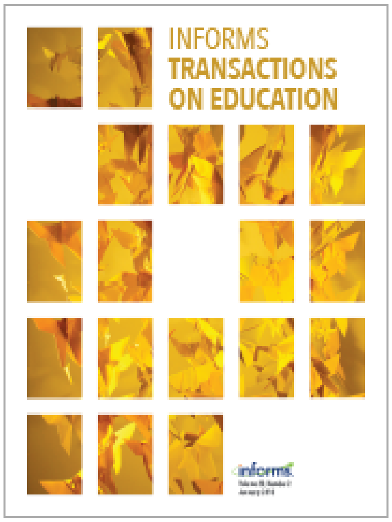


INFORMS Transactions on Education

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Puzzle—Shedding Light on Higher Dimensions

Martin J. Chlond

Vol. 15, No. 2, January 2015, pp. 197–199
ISSN 1532-0545 (online)
INFORMS Transactions on Education
http://dx.doi.org/10.1287/ited.2014.0135

Puzzle

Shedding Light on Higher Dimensions

Martin J. Chlond
Lancashire Business School, University of Central Lancashire, Preston PR1 2HE, United Kingdom, mchlond@uclan.ac.uk

Lights Out puzzles have been a popular computer pastime since the eighties. They consist of a grid of cells with each cell in one of two states, lit or unlit. When a cell is clicked, then that cell and its immediate neighbors toggle between lit and unlit. The objective is to identify the set of cells that must be clicked such that the entire grid is lit. Many variations on this theme devised by Jaap Scherphuis are at www.jaapsch.net/puzzles/lights.htm and one particularly interesting twist is at www.jaapsch.net/puzzles/java/lograph/lographapp.htm. This page implements, in the form of a Java applet, a set of tools to allow the creating, editing and playing of Lights Out puzzles on graphs. The nodes of the graph represent cells, and edges join any two cells that are considered neighbours. I show how to formulate Lights Out puzzles as integer programs and produce solutions using GLPK and Minisat. These are, respectively, open source software programs to solve Integer Programs and Satisfiability problems and are available within the GUSEK environment which may be downloaded from http://gusek.sourceforge.net/gusek.htm.

Keywords: puzzles; lights out; integer programming
History: Received: November 2014; accepted: November 2014.

Simple Graph Puzzle

Consider the simple planar graph in Figure 1.

In this puzzle, nodes 1 and 2 are initially lit and, for example, the set of nodes affected by a click on node 1 is $\{1, 2, 3, 4\}$. This graph, named "simple," may be downloaded and played using Jaap's applet. Full instructions on how to use the applet are also at the website.

To model this type of puzzle as an integer program we first define the set $N = 1..n$, where n is the number of nodes, and variables $x_i = 1$ if node i is clicked, 0 otherwise, $\forall i \in N$.

Also define set $A = \{A_1, A_2 \ldots\}$ whose elements are subsets of N. The subsets contain information about the connectivity of the graph. For example, $A_5 = \{2, 3, 5, 6\}$ indicates that node 5 will be affected by a click on any of these nodes.

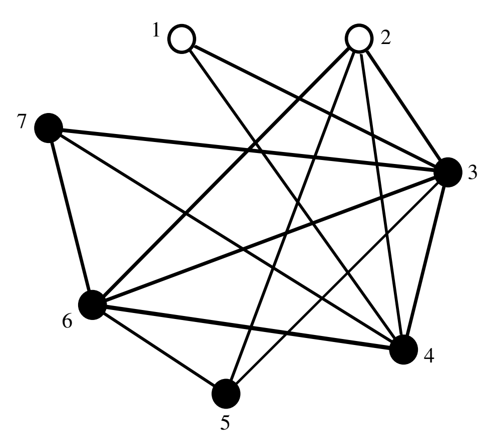
Finally, define dummy integer variables d_i, $\forall i \in N$.

The set notation allows us to express the necessary constraints as follows.

$$\sum_{j \in A_i} x_j = 2d_i + 1 \quad \forall i \in N$$

This will ensure that from a blank initial state all the nodes will be lit. The model may be modified to cope with different initial states by defining $init_i = 1$, if node i is already lit, 0 otherwise, $\forall i \in N$ and amending the right hand side of the constraints to become

Figure 1 Lights Out Puzzle on a Simple Graph

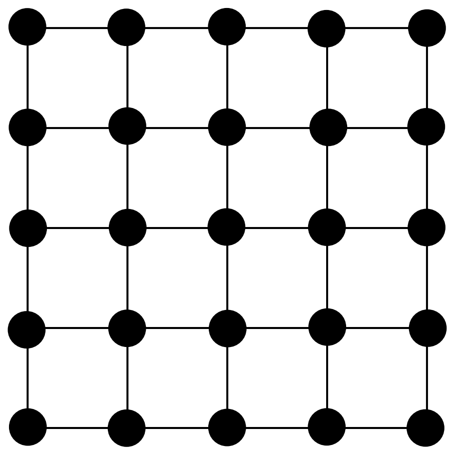


$$\sum_{j \in A_i} x_j = 2d_i + (1 - init_i) \quad \forall i \in N.$$

This causes a node that is lit in the initial state to be clicked an even number of times and hence remain lit and a node that is unlit in the initial state to be clicked an odd number of times.

A Mathprog model (lograph.mod) to solve this instance is included with this article (available as

Figure 2 5×5 **Puzzle**



supplemental material at http://dx.doi.org/10.1287/ited.2014.0135) and it may be modified to solve many similar puzzles.

5×5 Puzzle

A popular instance of a Lights Out puzzle is the 5×5 puzzle shown in Figure 2. This may be downloaded from the Jaap's website into the applet. The standard initial position is with all cells unlit.

This may be solved using the model for the simple graph puzzle by appropriate modification of the graph connectivity information but the special structure of the puzzle allows for a more convenient formulation where it is not required to explicitly describe the connectivity of every node.

Define the set $N = 1, \ldots, n$, where n is the size of a square grid, and variables $x_{i,j} = 1$ if cell $\{i, j\}$ is clicked, 0 otherwise, $\forall i \in N, j \in N$. Also define dummy integer variables $d_{i,j}$ and initial state $init_{i,j} = 1$, if cell $\{i, j\}$ is already lit, 0 otherwise, $\forall i \in N, j \in N$.

$$\sum_{\substack{k=i-1\\k\geq 1\\k\leq n}}^{i+1} x_{k,j} + \sum_{\substack{k=j-1\\k\geq 1\\k\leq n\\k\neq j}}^{j+1} x_{i,k} = 2d_{i,j} + (1 - init_{i,j}) \quad \forall i \in N, j \in N$$

The exclusion $k \neq j$ in the second summation is to avoid double counting $x_{i,j}$.

A Mathprog model (five.mod) is included with this article that will solve this instance and it may be modified to solve two dimensional grid problems in general.

Hypercube Puzzle

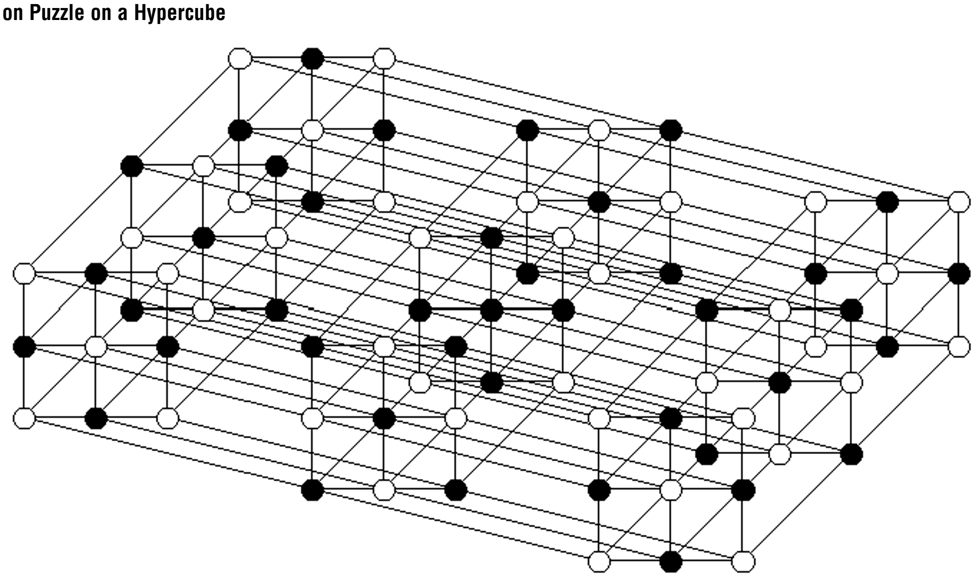
A graph of a hypercube grid of order 3 may be downloaded from Jaap's page and edited to produce the puzzle in Figure 3.

The constraints needed to model this puzzle may look a little unwieldy but they are a straightforward extension of those for a two dimensional grid.

$$\sum_{\substack{p=i-1\\p\geq 1\\p\leq n}}^{i+1} x_{p,j,k,m} + \sum_{\substack{q=j-1\\q\geq 1\\q\leq n\\q\neq j}}^{j+1} x_{i,q,k,m} + \sum_{\substack{r=k-1\\r\geq 1\\r\leq n\\r\neq k}}^{k+1} x_{i,j,r,m} + \sum_{\substack{s=m-1\\s\geq 1\\s\leq n\\s\neq n}}^{m+1} x_{i,j,k,s}$$
$$= 2d_{i,j,k,m} + (1 - init_{i,j,k,m}) \quad \forall i, j, k, m \in N$$

The implementation of this model turns out to be very slow running in GLPK and no solution was found after two hours. Fortunately, a small modification converts the model into an instance of the Boolean satisfiability (SAT) problem. For a brief discussion of the SAT problem and some examples of binary integer

Figure 3 **Lights on Puzzle on a Hypercube**



programs solved as SAT problems within the GUSEK environment see Chlond (2014). The dummy variables may be restructured as binary variables and the right-hand sides of the constraints replaced with

$$2d_{i,j,k,m,1} + 4d_{i,j,k,m,2} + (1 - init_{i,j,k,m}).$$

This modification is incorporated into lohypercube3 .mod included with this article and is solved within the GUSEK environment in under 20 seconds using the "–minisat" option. Lohypercube3.mod may be modified to solve hypercube grid puzzles in general. If the initial configuration of the puzzle contains symmetries, as the example does, then the solution time may be reduced further by eliminating symmetries from the search space. Constraints that achieve this are included in the model file and may be examined by the reader. These reduce the solution time to 6 seconds.

The hypercube3 puzzle was modified to represent a hyper-torus and this has also been uploaded to Jaap's site as hytpertorus3. The modeling of this is left as an exercise for the interested reader.

Supplemental Material

Supplemental material to this paper is available at http://dx.doi.org/10.1287/ited.2014.0135.

References

Chlond MJ (2014) IP satisfied. *INFORMS Trans. Ed.* 14(3):140–142.