Lights Out on a Random Graph

B. Forrest and N. Manno

Abstract - We consider the generalized game *Lights Out* played on a graph and investigate the following question: for a given positive integer n, what is the probability that a graph chosen uniformly at random from the set of graphs with n vertices yields a universally solvable game of *Lights Out*? When $n \leq 11$, we compute this probability exactly by determining if the game is universally solvable for each graph with n vertices. We approximate this probability for each positive integer n with $n \leq 100$ by applying a Monte Carlo method using 1,000,000 trials. We also perform the analogous computations for connected graphs.

Keywords : Lights Out; random graph; Monte Carlo

Mathematics Subject Classification (2020) **:** 05C57; 05C80; 60C99

1 Introduction

Lights Out is an electronic one-player game played on a 5×5 square grid of lights. Each one of the 25 square lights, when pressed, toggles itself and each of the lights with which it shares an edge. The goal in *Lights Out* is to produce a configuration in which all of the lights are turned off.

In this work, we consider *Lights Out* played on an undirected graph $\Gamma = (V, E)$, where there is one light for each element of V and pressing $v \in V$ toggles v and each vertex adjacent to v. We will call $C \subseteq V$ the *initial configuration* of a game of *Lights Out* on Γ where C is the set of vertices that are toggled on when the game begins. Winning *Lights Out* consists of finding a subset of V so that pressing the lights in that subset toggles exactly C; indeed it is only necessary to find a subset of V and not a sequence of that subset since the effect of pressing lights is independent of the order in which they are pressed. If such a subset of V exists, then we say that the game is *solvable*. Much is known about the solvability of particular initial configurations. For any graph Γ, if $C = V$ then the game is solvable, and if V is an $n \times n$ square grid with n odd and C is only the center square, then the game is solvable [6] [9] [15] [21]. We say that *Lights Out* on Γ is *universally solvable* if it is solvable for every initial configuration C. For example, the 3×3 square grid is universally solvable while the 5×5 grid is not [21].

In [1] and [2], Amin and Slater made substantial progress determining which graphs are universally solvable. In particular, they not only give equivalent conditions to determine when a graph is universally solvable but also classify the paths, spiders, and caterpillars that are universally solvable and provide a method to generate all universally solvable trees. The existence of so many classes of graphs that are universally solvable raises the primary question considered in this work: if a simple graph is chosen uniformly at random from the set of graphs with n vertices, what is the probability that it will yield a universally solvable game of *Lights Out*? Since games of *Lights Out* on disconnected graphs can be thought of as independent games on connected graphs, we also consider the analogous question for connected graphs. To approach this question, we implemented the algorithm for choosing a graph uniformly at random described by Dixon and Wilf in [7]. Our program was written in Java and our code is available at `http://github.com/nicolemanno/Lights-Out`.

This paper is organized as follows. In Section 2, we discuss the connection between *Lights Out* and linear algebra and give, for each $n \leq 11$, the number of graphs with n vertices that are universally solvable. We present the algorithm that we used to select a graph uniformly at random in Section 3 and give the results of our Monte Carlo experiments. In Section 4, we discuss possible extensions of our work and interesting open problems.

2 Lights Out and Linear Algebra

Lights Out can be studied through linear algebra by ordering the vertices of Γ. Once an order is chosen, subsets of the vertices of Γ can be represented by column vectors. Let Γ have n vertices and for simplicity we number the vertices from 1 to n. The subset C is given by the $n \times 1$ column vector

$$\overrightarrow{b} = \begin{bmatrix} b_1 & b_2 & \cdots & b_n \end{bmatrix}^T,$$

where $b_i = 1$ if vertex i is in C and is 0 otherwise.

The primary tool that we will use to study *Lights Out* is the *neighborhood matrix* of the graph Γ:

Definition 2.1 *Let $\Gamma = (V, E)$ be a graph with n vertices labeled 1 to n. The* neighborhood matrix $\mathcal{A} = [a_{i,j}]$ *of Γ is the $n \times n$ matrix in which $a_{i,j} = 1$ if $i = j$ or if the i-th and j-th vertices are adjacent and $a_{i,j} = 0$ otherwise.*

Remark 2.2 The matrix $\mathcal{A}$ is symmetric and the i-th row of $\mathcal{A}$ has a 1 in the j-th column if and only if pressing vertex i toggles vertex j. This matrix can also be expressed as the sum of the adjacency matrix of Γ and the $n \times n$ identity matrix.

When the column vector $\overrightarrow{b}$ represents the initial configuration of *Lights Out* on a graph Γ whose vertices are labeled 1 to n, solving the game reduces to finding the vector $\overrightarrow{x}$ so that $\mathcal{A}\overrightarrow{x} = \overrightarrow{b}$, where all of the coefficients are in the field $\mathbb{Z}_2$. The initial configuration represented by $\overrightarrow{b}$ is solvable if and only if it belongs to the column space of $\mathcal{A}$ [3]. This leads to the following fundamental observation about *Lights Out* on Γ: [11] [12] [21] [22]

Theorem 2.3 *Lights Out on a graph Γ is universally solvable if and only if the neighborhood matrix of Γ is invertible.*

When $\mathcal{A}$ is invertible, winning the game requires pressing the vertices represented by $\overrightarrow{x} = \mathcal{A}^{-1}\overrightarrow{b}$. Our program checks for invertibiliy by performing row reduction. Note that the order chosen for the vertices of Γ does not affect the invertibility of $\mathcal{A}$ since two different orderings give neighborhood matrices that are conjugate by permutation matrices.

Example 2.4 Figure 1 shows labeled representatives of the six distinct unlabeled connected graphs with four vertices. What follows are the neighborhood matrices corresponding to these labeled graphs.

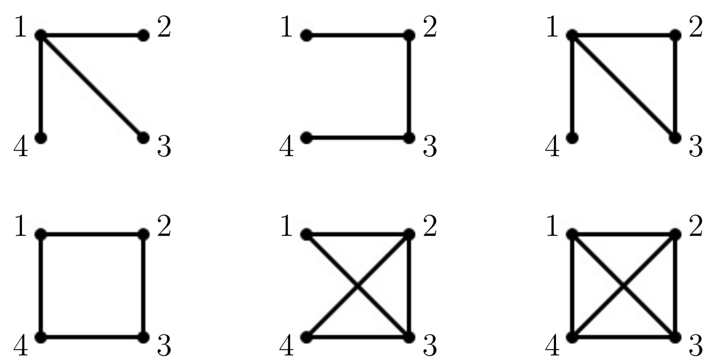

Figure 1: Labeled representatives of the six unlabeled connected graphs with 4 vertices.

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Row reducing each neighborhood matrix demonstrates that the only connected universally solvable graphs with 4 vertices are the four cycle and the length three path, represented in Figure 1 by the bottom left and top middle graphs, respectively.

Using the archives of graphs and connected graphs available from [17] and [18], we applied Theorem 2.3 to determine which of these graphs correspond to universally solvable games of *Lights Out*. For 1 to 10 vertices, the program took less than 5 minutes to run, while the run time to compute results for graphs with 11 vertices was approximately 10 hours; these computations were executed on a 2015 MacBook Pro wtih 16 GB of RAM. The results for all graphs follow in Table 1 while the results for connected graphs are in Table 2.

Table 1: For each n with $1 \leq n \leq 11$, the probability that a graph chosen uniformly at random from the set of graphs with n vertices is universally solvable.

Number of Vertices	Number of Graphs	Number of Universally Solvable Graphs	Probability Universally Solvable
1	1	1	1
2	2	1	0.5
3	4	2	0.5
4	11	4	0.363636
5	34	13	0.382353
6	156	47	0.301282
7	1044	339	0.324713
8	12346	4043	0.327474
9	274668	98375	0.358160
10	12005168	4553432	0.379289
11	1018997864	403286335	0.395768

3 Algorithm for Large Numbers of Vertices

When we consider graphs with more than 11 vertices, the methods of Section 2 are ineffective; for instance there are more than 10^{11} unlabeled graphs with 12 vertices. For this reason we applied a Monte Carlo method. For each n with $1 \leq n \leq 100$, we ran 1,000,000 trials of the experiment of selecting a graph uniformly at random from the set of graphs with n vertices and determined if each chosen graph produced a universally solvable game of *Lights Out*. To choose a graph uniformly at random, we followed the algorithm described by Dixon and Wilf in [7]. We will review their methods below.

3.1 Selection of a Graph Uniformly At Random

Consider the action of the symmetric group S_n on the set of all labeled graphs with n vertices where the action is given by permuting the labels. The unlabeled graphs with n vertices are in one-to-one correspondence with orbits of this action. For $g \in S_n$, let $\mathrm{Fix}(g)$

Table 2: For each n with $1 \le n \le 11$, the probability that a graph chosen uniformly at random from the set of connected graphs with n vertices is universally solvable.

Number of Vertices	Number of Connected Graphs	Universally Solvable Connected Graphs	Probability Universally Solvable
1	1	1	1
2	1	0	0
3	2	1	0.5
4	6	2	0.333333
5	21	9	0.428571
6	112	33	0.294643
7	853	290	0.339977
8	11117	3692	0.332104
9	261080	94280	0.361115
10	11716571	4454654	0.380201
11	1006700565	398728322	0.396074

be the set of fixed points of g under the action and consider the set $X = \{(g, \Gamma) \mid g \in S_n, \Gamma \in \text{Fix}(g)\}$. Dixon and Wilf observe that each orbit is represented the same number of times within X. By choosing an element from each conjugacy class of S_n, each orbit is also represented the same number of times within the subset of X corresponding to the chosen elements of S_n. So, we can choose a graph uniformly at random by

- first choosing a conjugacy class of S_n weighted by the product of the cardinality of its fixed point set and the number of elements in the conjugacy class,
- and then choosing a graph uniformly at random from the fixed point set of a representative of the chosen conjugacy class.

Note that the conjugacy classes of S_n are in one-to-one correspondence with partitions of n. More specifically, the partition $[k_1, k_2, \dots, k_n]$ with k_i parts of size i corresponds to the conjugacy class containing permutations with k_i cycles of length i for each i with $1 \le i \le n$. So, we label conjugacy classes by their corresponding partitions. Dixon and Wilf give the conjugacy class $[k_1, \dots, k_n]$ weight

$$w([k_1, ..., k_n]) = \frac{n! 2^{c(g)}}{\displaystyle\prod_{i=1}^{n} (i^{k_i} k_i!)},$$

where

$$c(g) = \frac{1}{2}\left(\sum_{i=1}^{n} l(i)^2\phi(i) - l(1) + l(2)\right), \qquad l(i) = \sum_{j=1,\, i|j}^{n} k_j,$$

and ϕ is the Euler phi function. Following their algorithm, we choose a partition π of n so that the probability of choosing $[k_1, \ldots, k_n]$ is equal to $w([k_1, \ldots, k_n])/n!g_n$, where g_n is the number of unlabeled graphs with n vertices. Our methods require knowledge of the value of g_n and this value has been computed up to $n = 140$ by Briggs [5] [20] using methods of Oberschelp [19]. While it is feasible to extend our results to $n = 140$, as n increases our approximations stabilize to the narrow range from 0.4185 to 0.4210 which limits the utility of extending the computation.

We follow the suggestion of Dixon and Wilf [7] and apply results of Oberschelp [19] to efficiently choose a partition. Within the set of partitions of n, the weight is concentrated in the partitions with the most parts of size 1. We choose a random number ξ, such that $0 \leq \xi < 1$, and compute the probabilities of the partitions in decreasing order of number of size 1 parts until the sum of the probabilities of the partitions that we have computed is greater than ξ. The final partition computed, the partition whose probability made the sum exceed ξ, is returned as π. The `random` method in the `RandomGraph` class implements this algorithm and is available at `http://github.com/nicolemanno/Lights-Out`. To create a partition of n with $n - k$ parts of size 1, we create a partition of k with no parts of size 1 and adjoin $n - k$ parts of size 1 to the partition. Since the partitions of n with more parts of size 1 usually have larger weight, k is typically small. In practice we are able to generate all of the partitions of k in decreasing order of the largest value in the partition, m, by recursively partitioning the remaining $k - m$. Within the code available at our github page, the implementation of this algorithm is the `uniquePartitions` method within the `Partition` class.

3.2 Choosing Graph From Fixed Point Set

Let P be the set of two element subsets of $\{1, 2, \ldots, n\}$. For a graph Γ with vertices labeled from 1 to n, the edges E of Γ are labeled as elements of P. Let $g \in \pi$ act on P by $g \cdot \{i, j\} = \{g(i), g(j)\}$. For $\Gamma \in \text{Fix}(g)$, each orbit in P under the action of g is either a subset of E or disjoint from E. Selecting a graph Γ uniformly at random from $\text{Fix}(g)$ reduces to computing the orbits in P under the action of g and constructing Γ by giving each orbit probability 0.5 of being in E. We compute the orbits in P under g by exhaustively applying g to each element of P. The code for this algorithm is available at our github page, and is implemented by the method `orbits` in the `RandomGraph` class.

3.3 Results

Following the procedure given by Dixon and Wilf in [7], our algorithms to select a partition π and graph Γ, from Subsections 3.1 and 3.2, respectively, produce a representative labeled graph Γ of an unlabeled graph chosen uniformly from the set of graphs with n

Table 3: For each n with $1 \leq n \leq 27$, the probability that a graph chosen uniformly at random from the set of graphs with n vertices is universally solvable, approximated by 1,000,000 trials. For $n \geq 11$, the margin of error at 95% confidence is less than 0.001.

Number of Vertices	Probability Connected	Probability Universally Solvable	Number of Vertices	Probability Connected	Probability Universally Solvable
1	1	1	15	0.999083	0.417295
2	0.500186	0.499814	16	0.999465	0.418368
3	0.501065	0.498932	17	0.999744	0.417804
4	0.545260	0.362558	18	0.999859	0.419841
5	0.617609	0.382157	19	0.999926	0.419059
6	0.717691	0.301979	20	0.999974	0.419294
7	0.817183	0.324418	21	0.999984	0.420138
8	0.900342	0.328411	22	0.999988	0.419788
9	0.950290	0.357775	23	0.999994	0.419580
10	0.975863	0.378710	24	0.999999	0.419234
11	0.987933	0.395842	25	1	0.419309
12	0.993693	0.405955	26	0.999998	0.419638
13	0.996637	0.411057	27	0.999999	0.419374
14	0.998301	0.414513			

vertices. As discussed in Section 2, we check if Γ is universally solvable by row reducing its neighborhood matrix.

For each n from 1 to 27, we ran 1,000,000 experiments where we chose a graph with n vertices uniformly at random and determined if it was universally solvable. We also computed the probability of a graph with n vertices being connected when chosen uniformly at random. These results are shown in Table 3. For those same values of n, we also ran 1,000,000 experiments where we chose a connected graph with n vertices uniformly at random and determined if it was universally solvable. We accomplished this by choosing a graph with n vertices uniformly at random and re-running the experiment if the chosen graph was not connected. We checked if the graph is connected by starting at node 1 and performing a depth first traversal of the spanning tree of the graph made by visiting the previously unvisited adjacent vertex with smallest numerical label. If every vertex in the graph had been visited by the end of the traversal, then the graph was connected. Within the code available at our github page, the `traverse` and `isConnected` methods in the `MatrixGenerator` class implement this algortihm. Our results for connected graphs with n from 1 to 27 are shown in Table 4. For each n from 28 to 100, we did not handle the connected case separately. As n approaches ∞, the probability that the chosen graph

Table 4: For each n with $1 \leq n \leq 27$, the probability that a graph chosen uniformly at random from the set of connected graphs with n vertices is universally solvable, approximated by 1,000,000 trials. For $n \geq 11$, the margin of error at 95% confidence is less than 0.001.

Number of Vertices	Probability Universally Solvable	Number of Vertices	Probability Universally Solvable
1	1	15	0.417291
2	0	16	0.418391
3	0 .499886	17	0.417805
4	0.332844	18	0.419837
5	0.428600	19	0.419054
6	0.294965	20	0.419296
7	0.340211	21	0.420136
8	0.333021	22	0.419794
9	0.360898	23	0.419580
10	0.379616	24	0.419235
11	0.396218	25	0.419309
12	0.406112	26	0.419638
13	0.411093	27	0.419373
14	0.414537		

is connected approaches 1. For each of these values of n, all 1,000,000 graphs chosen uniformly at random were connected, and we determined if each of the 1,000,000 chosen graphs was universally solvable. We present our results for these values in Table 5.

For values of $n \geq 11$, our sample size is small relative to the set of all graphs with n vertices; we can assume that our experiment produces a binomial distribution giving a margin of error of less than 0.001 at 95% confidence. As confirmation of the correctness of our algorithms and implementation, observe that for values of n from 1 to 11, our results in Tables 3 and 4 closely match the results in Section 2. Additionally, for each n from 1 to 27, the probability that a graph chosen uniformly at random from the set of graphs with n vertices is connected, as given in Table 3, agrees with known values [5].

The runtime for the results displayed in Tables 3 and 4 was 30 minutes, while it took 48 hours to compute the results for Table 5. These computations were performed on a laptop with a 2.80 GHz processor and 16 GB of RAM running Windows 11.

Table 5: For each n with $28 \leq n \leq 100$, the probability that a graph chosen uniformly at random from the set of graphs with n vertices is universally solvable, approximated by 1,000,000 trials; all graphs chosen were incidentally connected. For each n, the margin of error at 95% confidence is less than 0.001.

Number of Vertices	Probability Universally Solvable	Number of Vertices	Probability Universally Solvable	Number of Vertices	Probability Universally Solvable
28	0.419273	53	0.419991	78	0.419690
29	0.419329	54	0.419592	79	0.418720
30	0.419358	55	0.420166	80	0.419922
31	0.419409	56	0.420290	81	0.419727
32	0.419396	57	0.418865	82	0.419339
33	0.418807	58	0.419681	83	0.419351
34	0.419929	59	0.419318	84	0.420456
35	0.418619	60	0.419115	85	0.419437
36	0.419861	61	0.418608	86	0.419870
37	0.420124	62	0.418717	87	0.419644
38	0.419721	63	0.419397	88	0.418764
39	0.419141	64	0.419275	89	0.419063
40	0.420303	65	0.419000	90	0.419413
41	0.419691	66	0.419709	91	0.418826
42	0.419005	67	0.419992	92	0.419806
43	0.419342	68	0.418893	93	0.420023
44	0.419507	69	0.419053	94	0.420860
45	0.419345	70	0.420118	95	0.420415
46	0.420292	71	0.419159	96	0.418757
47	0.419140	72	0.419537	97	0.419563
48	0.418924	73	0.418961	98	0.419010
49	0.418542	74	0.419708	99	0.419510
50	0.418690	75	0.419446	100	0.419362
51	0.419030	76	0.419571		
52	0.419023	77	0.419299		

4 Future Work

There are a number of potential extensions of this work. Variants of *Lights Out* with more than one toggle mode have been considered by many authors [4] [8] [10] [14] [16] [23]. It is possible to extend our results in this direction.

Instead of choosing a graph uniformly at random from the set graphs with n vertices, one could restrict the set of graphs under consideration to those with m edges and n vertices. Fixing n and examining how the probability of choosing a universally solvable graph changes as m changes is an interesting direction to extend this work.

Our work suggests that as n approaches ∞, the probability that a graph with n vertices chosen uniformly at random is universally solvable is approximately 0.419. This is only an approximation and conjecture, investigating this is an intriguing open problem.

Acknowledgments

The authors acknowledge the contributions of both the referee and the editor. Their insightful suggestions and helpful comments improved this work significantly.

References

[1] A.T. Amin, P.J. Slater, Neighborhood domination with parity restrictions in graphs, *Congr. Numer.*, **91** (1992), 19–30. MR

[2] A.T. Amin, P.J. Slater, All parity realizable trees, *J. Combin. Math. Combin. Comput.*, **20** (1996), 53–63. MR

[3] M. Anderson, T. Feil, Turning lights out with linear algebra, *Math. Mag.*, **71** (1998), 300–303. MR

[4] L. Ballard, E. Budge, D. Stephenson, Lights out for graphs related to one another by constructions, *Involve*, **12** (2019), 181–201. MR

[5] K. Briggs, Combinatorial Graph Theory, (2022, May 11). Retrieved from `http://keithbriggs.info/cgt.html`

[6] R. Cowen, S.H. Hechler, J.W. Kennedy, A. Ryba, Inversion and neighborhood inversion in graphs, *Graph Theory Notes N.Y.*, **37** (1999), 37–41. MR

[7] J.D. Dixon, H.S. Wilf, The random selection of unlabeled graphs, *J. Algorithms*, **4** (1983), 205–213. MR

[8] S. Edwards, V. Elandt, N. James, K. Johnson, Z. Mitchell, D. Stephenson, Lights Out on finite graphs, *Involve*, **3** (2010), 17–32. MR

[9] H. Eriksson, K. Eriksson, J. Sjöstrand, Note on the lamp lighting problem, *Adv. Appl. Math.*, **27** (2004), 357–366 MR

[10] A. Giffen, D.B. Parker, On generalizing the *Lights Out* game and a generalization of parity domination, *Ars Combin.*, **111** (2013), 273–288. MR

[11] J. Goldwasser, W. Klostermeyer, G. Trapp, Characterizing switch-setting problems, *Linear Multilinear Algebra*, **43** (1997), 121–136. MR

[12] J. Goldwasser, W. Klostermeyer, G. Trapp, C. Zhang, *Setting switches in a grid, Technical Report TR-95-20*, Department of Statistics and Computer Science, West Virginia University, 1995.

[13] C. Gray, L. Mitchell, M. Roughan, Generating connected random graphs, *J. Complex Netw.*, **7** (2019), 896–912. MR

[14] S. Gravier, M. Mhalla, E. Tannier, On a modular domination game, *Theoret. Comput. Sci.*, **306** (2003), 291–303. MR

[15] Y. Hayata, M. Yamagishi, On weight-one solvable configurations of Lights Out Puzzle, *Involve*, **12** (2019), 713–720. MR

[16] M. Hunziker, A. Machiavelo, J. Park, Chebyshev polynomials over finite fields and reversibility of σ-automata on square grids, *Theoret. Comput. Sci.*, **320** (2004), 465–483. MR

[17] B.D. McKay, Combinatorial Data, available online at the URL: `http://users.cecs.anu.edu.au/~bdm/data/graphs.html`

[18] B.D. McKay, A. Piperno, Practical Graph Isomorphism, II, *J. Symbolic Comput.*, **60** (2013), 94–112. MR

[19] W. Oberschelp, Kombinatorische Anzahlbestimmungen in Relationen, *Math. Ann.*, **174** (1967), 53–78. MR

[20] N.J. Sloane, A000088, available online at the URL: `http://oeis.org/A000088`

[21] K. Sutner, Linear cellular automata and the Garden-of-Eden, *Math. Intelligencer*, **11** (1989), 49–53. MR

[22] K. Sutner, The σ-game and cellular automata, *Amer. Math. Monthly*, **97** (1990), 24–34. MR

[23] M. Zaidenberg, Periodic harmonic functions on lattices and points count in positive characteristic, *Cent. Eur. J. Math.*, **7** (2009), 365–381. MR

Bradley Forrest
Stockton University
101 Vera King Farris Drive
Galloway, NJ 08205
E-mail: `bradley.forrest@stockton.edu`

Nicole Manno
E-mail: `nicoleannemanno@gmail.com`

Received: June 24, 2021 **Accepted**: July 26, 2022
Communicated by Noah Williams