

# 点灯问题的解法

吴先强, 李 睿

(华东交通大学 电气与电子工程学院, 江西 南昌 330013)

**摘要:** 在对点灯游戏进行深入分析的基础上, 建立了形式化的数学模型, 由不同的解决思路, 设计了几种相应的算法, 利用 MATLAB 编程, 完整的解决了此问题; 分析比较了几种算法的效率, 给出了一些有用结论; 并且对该问题的可解性进行了探讨.

**关键词:** 点灯问题; 数学建模; MATLAB 编程; 算法分析

**中图分类号:** TN911.3

**文献标识码:** A

点灯游戏是一个十分有趣的智力游戏, 它的规则是: 有  $N$  行  $N$  列 ( $N$  为正整数) 的灯, 游戏开始时的初始状态为全部灯是灭的, 当你点击其中一盏灯 (点灯), 被点的灯和它的紧邻上下左右 (对于位于中间的灯, 它的紧邻上下左右共有 4 盏灯, 角上的灯的紧邻只有 2 盏灯, 边上非角上的灯的紧邻为 3 盏灯) 的灯的状态将全部改变, 原来是灭的灯变亮, 原来亮的灯则变灭, 要求游戏玩家设法将全部的灯点亮<sup>[1]</sup>.

该游戏更一般的玩法是: 初始局面随机确定一些灯是亮的, 另外的是灭的, 按照上面同样的点灯规则, 要求玩家设法将全部的灯点亮. 该游戏还有多个变种<sup>[2]</sup>, 本质是一样的.

## 1 问题分析与建模

这个问题看似简单, 实际玩的时候, 并不容易. 问题在于每盏灯的亮或灭与紧邻的灯的操作相关, 这种相关性通过紧邻传播, 使得所有灯的状态相互制约. 下面引入一些定义, 进一步清楚地描述问题, 通过分析得出一些有用结论, 并在此基础上设计算法, 编制程序, 用计算机来解决此问题.

**局面:** 灯的亮与灭的一种分布状态. 可以用一个

$N$  行  $N$  列的 0-1 矩阵 (二维数组)  $A$  描述,  $a_{ij}$  表示第  $i$  行第  $j$  列的灯灭 (亮).

**初始局面:** 游戏开始时的局面. 为全 0 矩阵或 0 与 1 随机分布的矩阵.

**目标局面:** 要求通过点灯操作达到的一种灯的亮与灭的分布状态. 一般为一个全 1 矩阵.

**点灯方案:** 游戏玩家实施的一个点灯操作序列. 可用  $[a_{i1,j1}, a_{i2,j2}, \dots, a_{in,jn}]$  表示.

**有解局面:** 一个局面如果存在点灯方案, 使得该局面转变为目标局面则称为有解局面, 否则称为无解局面.

**等价局面:** 如果一个局面存在某个点灯方案, 可以转变为另一个局面, 则两个局面称为等价局面.

**可行方案:** 能够使初始局面转变为目标局面的一个点灯方案.

**简单方案:** 一个可行方案称为简单方案, 如果在这个方案里, 任意一个灯被操作的次数最多为 1.

**等价方案类:** 由一个简单方案派生出来的一系列可行方案, 具体解释见后面的推论 1.

**最简方案:** 所有方案里, 操作次数最少的可行方案.

点灯问题就是要求找出可行方案, 使初始局面转变为目标局面.

在上述基础上,给出如下一些结论:

**结论 1** 对于  $N$  行  $N$  列的点灯问题,可能出现的局面数为  $2^{N \times N}$ ,考虑到对称性,实质上不同的局面数为  $2^{N \times N}/4 = 2^{N \times N - 2}$ . 理论上可能出现的初始局面数也是上述数字.

**结论 2** 点灯方案里点灯的顺序无关重要. 也就是说如果存在某个可行方案,那么任意改变点灯的次序,所得的新方案仍然是一个可行方案.

理由是:最后一盏灯是亮还是灭由自身与紧邻上下左右的总的操作次数决定,而每次点灯操作的效果都是独立的,与之前或之后的操作不存在相互影响. 故结论 2 成立.

**结论 3** 对某一盏灯的操作,操作偶数次与操作 0 次效果相同,操作奇数次与操作 1 次相同. 由操作规则易知结论显然成立.

**推论 1**  $N$  行  $N$  列的点灯问题的一个可行方案可以表达为一个  $N$  行  $N$  列的 0-1 矩阵  $B, b_{ij}$  表示对第  $i$  行第  $j$  列操作了 0 (或 1) 次.

由结论 2 可知点灯次序不重要,可用矩阵表达点灯方案,而由结论 3 可知对每盏灯的操作次数最多为 1 即可. 故推论 1 成立.

这样,点灯问题的求解就是寻找一个合适的 0-1 矩阵. 这样得到的 0-1 矩阵显然就是一个简单方案. 在一个简单方案的基础上,任意调整点灯顺序,对任意一盏灯增加任意偶数次操作所得到的方案的总体,称为一个等价方案类.

**结论 4** 一个点灯问题的初始局面为可解局面,当且仅当初始局面和目标局面为等价局面. 这个结论由前面的定义很容易看出.

通过上面的分析,建立点灯问题的形式化模型如下:

对于一个随机给定的  $N$  行  $N$  列 0-1 矩阵  $A$ , 要求一个相同维数的 0-1 矩阵  $B$ , 对  $a_{ij}$  进行  $b_{ij}$  次操作,操作规则是对  $a_{ij}$  及其紧邻取反,使得  $A$  变为一个全 1 矩阵.

一个初始局面到底是不是可解局面,一个可解的初始局面到底有多少个简单方案,开始时是难以看出的. 最简方案也不一定容易得出. 下面通过分析,用计算机来解决这些问题.

## 2 算法分析与程序设计

### 2.1 算法 1

最直接最简单的想法是穷举  $B$  的所有可能取

值,一一判断它们是否是可行方案. 判定一个方案可行的依据是  $a_{ij} + b_{ij} + b_{i-1,j} + b_{i,j-1} + b_{i,j+1} + b_{i+1,j}$  为一奇数对每个  $i, j$  均成立. (为了得到统一简洁的表达,把局面矩阵,方案矩阵等都扩展为周围用 0 包围的矩阵.) 得到如下的算法 1.

S1 读入一个随机的初始局面,为了便于后面处理,扩展该矩阵,周围用 0 包围;

S2 循环,对每个可能的方案矩阵,检验是否为简单方案;

$\text{mod}(a_{ij} + b_{ij} + b_{i-1,j} + b_{i,j-1} + b_{i,j+1} + b_{i+1,j}, 2) = 1$ ;

S3 输出简单方案的信息,无简单方案的话给出提示,有则输出所有简单方案,统计个数.

为了比较分析算法的效率,记录程序运行的时间,用 MATLAB 编程便于处理矩阵,语句简洁紧凑,记录结果可以用三维数组,用 tic, toc 语句很容易记录程序运行时间<sup>[2]</sup>. 综合上述特点,采用 MATLAB 语言编写程序.

采用此方法的最大缺点是要对每个可能的简单方案进行检验,算法的时间复杂度为  $O(2^{N \times N})$ ,随着问题规模的扩大,该算法的效率非常低. 在作者的计算机(奔腾 4 CPU, 512M 内存)上面,解决 5 行 5 列的初始局面为全灭的点灯问题花时 874 秒.

### 2.2 算法 2

仔细分析算法 1, 容易发现其中做了很多不必要的搜索判断. 比如,  $b_{1,1}, b_{1,2}, b_{2,1}$  确定后,就可以知道最后第 1 行第 1 列的灯的最终状态了,如果是灭的,显然没有必要再往下试探. 根据这个想法,对算法 1 做相应改写,得到算法 2.

S1 同算法 1;

S2 循环,但是在试探的过程中边试探边判断能对最终状态做出判定了的灯的状态;

S3 同算法 1.

经过上述改进后,算法效率有了较大的提高. 同样解决 5 行 5 列全灭初始局面的点灯问题,用时仅为 0.64 秒. 该算法缺点是代码繁复,问题规模扩大时,程序改写复杂,时间复杂度还是没有明显的减小.

### 2.3 算法 3

仔细分析可知:当操作方案的第 1 行确定后,为了保证第 1 行的灯最终为亮的状态,第 2 行应进行的操作很容易确定. 同样,第 1 行和第 2 行都确定后,为了保证第 2 行的灯最终全部为亮,第 3 行的操作很容易确定. 依次类推,只要第 1 行操作一确定,整个操作方案就很容易确定了. 因此只要对可能简

单方案的第1行进行枚举即可.由此得算法3:

S1 同算法1;

S2 对简单方案的第1行枚举,其余行由第1行元素递推得到.然后判断此方案是否为可行的简单方案;

S3 同算法1.

程序如下:

```
function [B,k] = diandeng( A)
tic;
sizea = size( A );
n = sizea( 1 ); % 计算输入局面的阶;
a = zeros( n + 2 );
a( 2: n + 1, 2: n + 1 ) = A; % 在初始局面的周围
% 加一圈0,以便使用统一的公式由第一行的数字
% 递推下面的所有数字;
b0 = ones( n ); % 用来表示全亮局面;
bb = zeros( n + 2 ); % 用来记录操作方案,
% 扩展了的 12* 12 的;
aa = zeros( n + 2 ); % a 经过试探矩阵 bb 的操作后
% 形成的矩阵,初始化为全0 矩阵;
k = 1; % 记录简单方案的个数
for q = 0: 2^n - 1
    bb( 2, 2: n + 1 ) = binary( q, n );
    for ii = 3: n + 1
        for jj = 2: n + 1
            bb( ii, jj ) = mod( a( ii - 1, jj ) + bb( ii - 2, jj ) ...
                + bb( ii - 1, jj - 1 ) + bb( ii - 1, jj ) ...
                + bb( ii - 1, jj + 1 ) + 1, 2 );
        end
    end
    b = bb( 2: n + 1, 2: n + 1 );
    for l = 2: n + 1
```

```
        for r = 2: n + 1
            aa( l, r ) = mod( a( l, r ) + bb( l, r ) + bb( l - 1, ...r ) + bb( l
                + 1, r ) + bb( l, r - 1 ) + bb( l, r + 1 ), 2 );
        end
    end
    if sum( sum( aa ) ) == n^2 % aa 为全1 矩阵,
    % 则元素和为 n^2,代表点灯成功;
        B( :, k ) = b;
        k = k + 1;
    end
end
k = k - 1;
fprintf( '共有 %g 个可行方案. \n', k );
if k >= 1
    disp( '可行方案如下: ' );
    disp( B );
else
    B = I;
end
toc;
```

其中 binary 函数用于把一个10进制的数转变为对应的二进制数组,定义如下:

```
function y = binary( x, n)
for t = 1: n
    y( n + 1 - t ) = mod( x, 2 );
    x = ( x - mod( x, 2 ) ) / 2;
end
```

经过改进后,算法的时间复杂度为  $O(2^N)$ . 相比前面的算法,时间复杂度大大降低.例如解决如图1问题仅用时0.063秒.(图2为图1问题的矩阵编码.图3为可行方案的编码.)

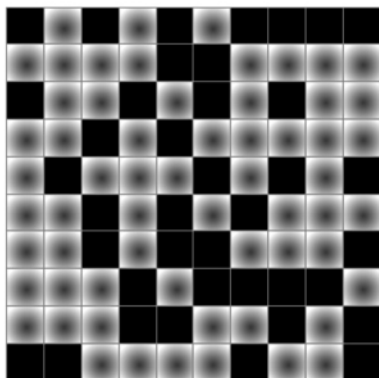


图1 初始局面

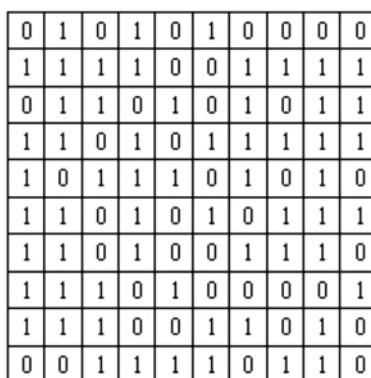


图2 初始局面的0-1矩阵表示

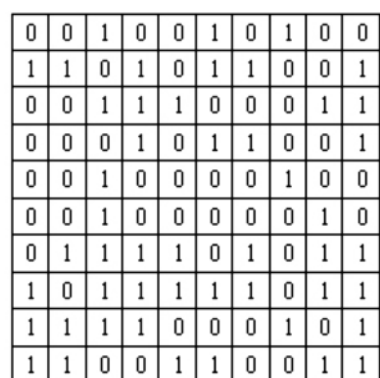


图3 可行简单方案

### 3 可解性分析

利用上述程序可以对点灯问题的可解性做进一步分析,下面对  $10 \times 10$  做分析.

**引理** 对任意的局面,必存在一个等价局面,该局面至多只剩最后一行存在不亮的灯.

理由是对于一般的局面,我们可以只操作第二行,使第一行的灯全亮,接着可以操作第三行使第二行的灯全亮,依此类推,可以很简单的使除最后一行的灯全亮.

由此可以得出一个结论: 相互不能转化的局面至多只有  $2^n$  个. 用下面的程序可以分析这些情况.

```
n = input(' n = ');
a = ones( n );
ii = 0;
for m = 0: 2^n - 1
    a( 1, :) = binary( m, n );
    [B, k] = diandengfun( a );
    ii = ii + 1;
    fanum( ii ) = k;
end
```

令  $n = 10$ , 很容易得出一个结论: 对于  $10$  阶而言,任意的初始局面都是有解的.

对于别的  $n$  而言则不一定. 这可能正是  $10 \times 10$  的游戏玩得最多的原因.

### 4 结语

(1) 对点灯小游戏深入分析,建立数学模型,更清楚地看到了问题的本质;

(2) 仔细分析问题,设计一个合适的算法对解决问题非常关键. 选择 MATLAB 解决此问题也是非常方便的.

按照算法 3 编制的程序能在极短的时间内解决网络上常见的点灯小游戏问题.

### 参考文献:

- [1] Strongcom. 点灯游戏随机版 [EB/OL]. <http://game.23kk.com/sub/940/>, 2005-10-20/2007-12-01.
- [2] 张志涌,徐彦琴,等. MATLAB 教程 [M]. 北京: 北京航空航天大学出版社, 2003.

## Solution to the Problem of Lighting Lamps

WU Xian - qiang, LI Rui

( School of Electrical and Electronic Engineering, East China Jiaotong University, Nanchang 330013, China )

**Abstract:** On the basis of deeply analyzing lighting in the game, a formal mathematical model is established, several algorithms with various solutions are designed, and a complete solution through MATLAB programming is given. The paper also compares efficiency of several algorithms, and gives some useful conclusions, discussing its solutions.

**Key words:** lighting lamps; mathematical modeling; MATLAB programming; analysis of algorithms

( 责任编辑: 刘棉玲 )