

数学建模讲座:元胞自动机

aqua2001

数学中国超级版主

September 4, 2010

大家好,我是数学中国超级版主 aqua2001。我们这次做元胞自动机的讲座。

元胞自动机是一个在数学建模中有用的工具。在这里,我们打算通过一些模型,来初步介绍元胞自动机的应用。使用计算机进行模拟往往在建模过程中有很大的用途,但凭空说“模拟真实世界”往往让人觉得难以下手。而元胞自动机则往往能给模拟方法提供一个容易思考的框架。

考虑到大家在建模中的实用性,所以在介绍当中,我会尽量多做直观的介绍,尽量避免数学计算。详细的数学内容大家可以参照相关书籍。讲解中有一些图片或说法来源于书籍或网络。

我们在讲解当中,除了一些直接用到元胞自动机的模型,也介绍了在元胞自动机的发展过程当中一些比较有影响力的内容。它们不一定对建模有直接的用途,但我希望大家对它的了解能够更广泛一些,这样可能对它进行更灵活地运用。

元胞自动机(Cellular Automata 或 Cellular Automaton, CA)是空间和时间都离散,物理参量只取有限数值集的物理系统的理想化模型。

它的提出,最早是冯·诺依曼在研究能够自我复制的自动机时提出来的。其特点是,空间被分成离散的格子(可以是方形、三角形或六边形等),称之为元胞(Cell)。元胞处于若干可能的状态之一,而且随着时间,其状态可以演化。每个元胞状态的演化,往往要受到临近元胞的状态的影响。而且,在传统的元胞自动机中,每个元胞的变化都是同时进行的。

最著名的元胞自动机的例子,应属 Conway 提出的“生命游戏”。它的提出并不是为了具体的应用,但其蕴含的丰富内容引起了许多人的兴趣。假设二维空间被划分成方形的格子,当然每个格子都是一个元胞。假设每个元胞只处于两种状态之一:死的与活的。死的可以记为 0,活的可以记为 1。

现在我们来考察每个元胞周围的“邻居”。“邻居”的定义当然并不唯一,

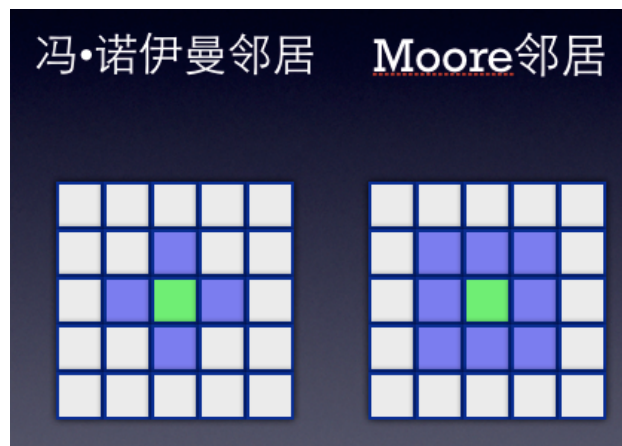


Figure 1: 标准邻居

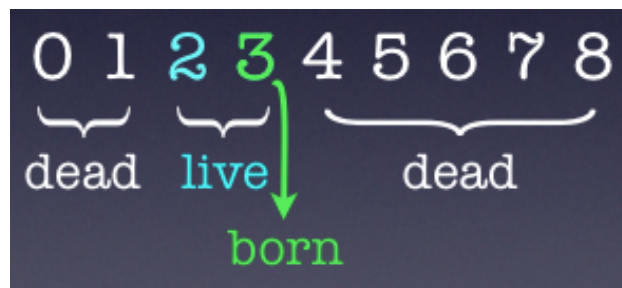


Figure 2: 死亡,存活和重生的规则

有时考虑的是它的上下左右的邻居(如果是三维空间,则还有前后),这被称为“冯·诺依曼邻居”;有时考虑的是包围在它周围的全部邻居,被称为 **Moore** 邻居。在更高维的空间也可以有类似的定义。当然,也完全可以考虑其它类型的邻居,也可以包括更广的范围。

在生命游戏中,考察每个元胞的 **Moore** 邻居。如果邻居中只有 1 个或者没有活的元胞,或者有 4 个以上活的元胞,那么这个元胞如果是活的,就变成死的(由于孤独或拥挤),如果本来是死的就不用变了。若邻居中有 2 或 3 个活的元胞,那么如果这个元胞是活的,可以继续存活。如果邻居中有 3 个活的元胞,而中间的元胞现在是死的,那么可以再生。

整个空间的初始的状态可以人为设计,也可以随机设定。随着时间的推移,每个元胞都或死或生,然而空间的整体却出现了非常复杂的状态演化。

有一些活元胞组成的形状会稳定不动(例如四个活元胞排成田字形),而有一些则会周期性变化(例如三个活元胞排成一字形)。有一些活的元胞可

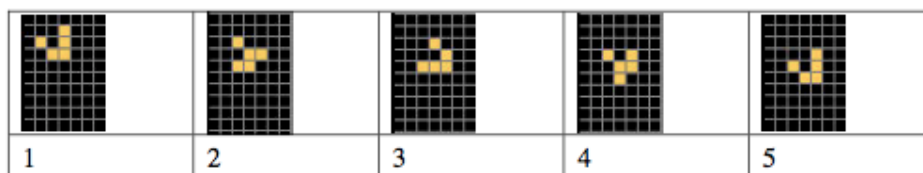


Figure 3: 滑翔机的运动

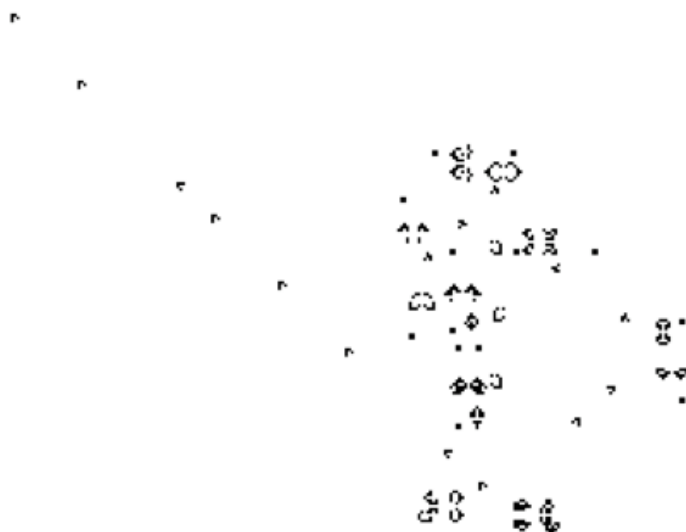


Figure 4: 滑翔机发射器

以排列成能够整体运动的形态(称之为 **Slider**, 我们暂时翻译成滑翔机。)

有一些元胞的组合可以在自己的周期变化当中不断地发射出滑翔机, 有一些可以在自己的周期当中“吃掉”滑翔机。更多的则是体现出复杂而难以捉摸的演化状态。活元胞的总数随时间的变化, 总的来说也无规律可循。

很奇妙的事情是, 有一些组合的形状, 可以不停地发射出滑翔机, 而发射的滑翔机之间的间隔, 竟然是非周期的, 呈现出貌似随机的样子。

我们来看看常见的一些样子, 见图5:

Conway 证明, 生命游戏事实上等价于一个通用图灵机。通过选择不同的初始条件, 可以完成一切计算机能够完成的算法演算。在这个证明的思路当中, “传输信息”的工作, 便是用滑翔机来进行的。

这个结果, 事实上意味着生命游戏可以产生几乎无数的复杂现象。对它也有许多有趣的研究。

对元胞自动机的系统研究, 其实还是 **Wolfram** 在 1983 年开始进行的。

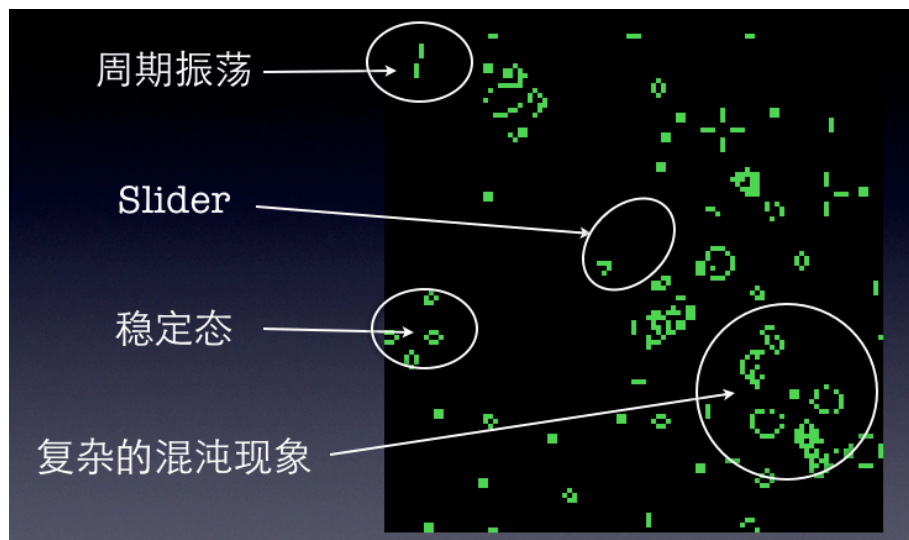


Figure 5: 生命游戏的常见场景



Figure 6: 演化规则

Wolfram 就是著名软件 Mathematica 和 Wolfram 公司的创始人。

他开始的时候,研究的是最基本的一维情况。也就是空间是一维的,也就是所有元胞排列成一维的链条。

每个元胞 i 具有两种可能的状态, 0 或 1。而 i 的状态变化取决于它的左右两个“邻居”。我们用一个图来描述演化的规则:

中间的元胞将随着自己和左右情况的不同,而演化成 a_7, \dots, a_0 等不同状态。把 a_7 到 a_0 都指定一个 0 或 1 的数字,也就确定了一套完整的演化规则。我们等于是用穷举法,把在一切情况下的演化结果,全部列举清楚了。

我们还可以用图示的方法,举一个演化规则的例子。

这个图中,每种情况都指定好了演化的结果。用图来代表,绿色代表 1,黑色代表 0。把这个数字“10000111”看成二进制的数字,化成十进制其实是 135。所以我们可以把这个规则命名为规则 135。

对一个初始状态来说,使用规则 135 来演化,随着时间,会变成什么样子?

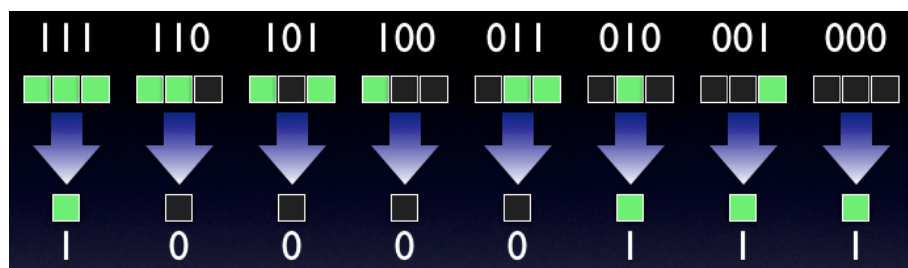


Figure 7: 135 号演化规则

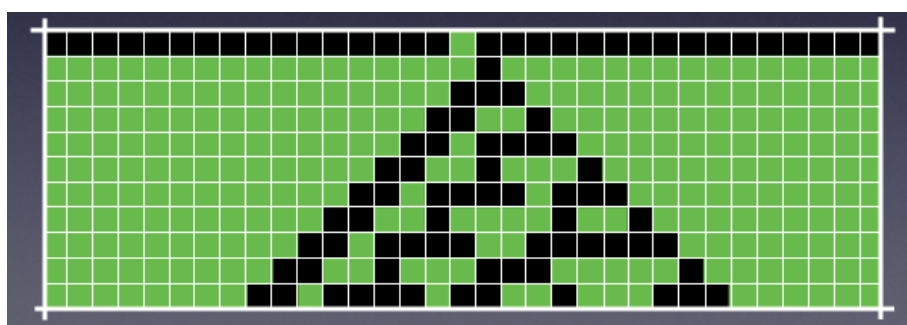


Figure 8: 135 号演化规则的进程

这是很容易用计算机程序验证的。我们可以把每一个时间步得到的结果画在前一步的下面,可以看到一些有趣的图案。

规则 13(你可以自己把 13 化成二进制,填在刚才讲“规则”的那个示意图里,看看这个规则具体说的是什么)从一个随机的初始状态开始,逐渐演化成一个稳定的状态而不再变化,图??体现出了。

图中没有画出元胞的方格边界。显明的竖线,是由于那个元胞一直处于 0 或 1,而产生的现象。

规则 56 则有所不同。它可由图10来反映:

看似斜线的图样,意味着每过一个时间步,处于 1 的元胞的位置,向右推移一格(其实处于 0 的元胞的位置,也同样向右推移了一格)。如果空间是无限的,那这就类似一种“匀速运动”,而如果空间的右侧和左侧连成一个环,那这可以看成是一种周期性的演化。

规则 18 则产生了比较奇怪的图案:

这看起来像分形几何里的图案。用动力系统的术语来说,这种演化,很类似混沌的状态。如果从只有一个 1 的初始状态开始,结果就更清晰一些,请看图12:

这非常类似 Sierpinski 三角形。

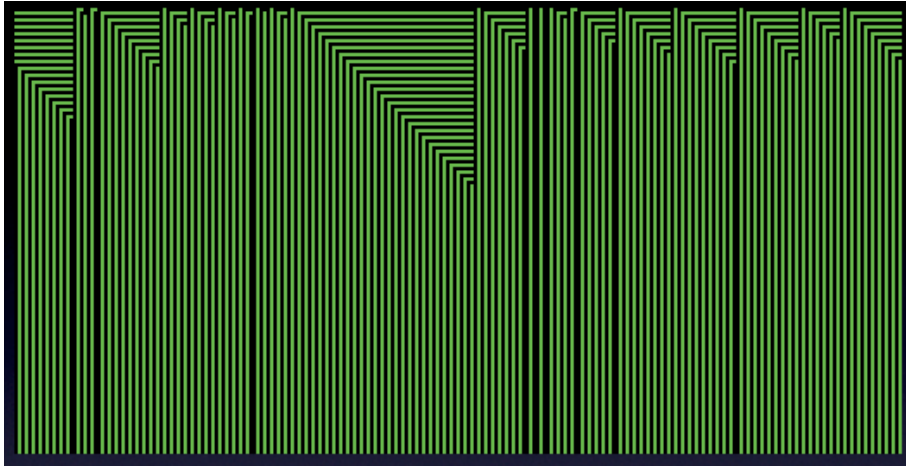


Figure 9: 13 号演化规则的进程

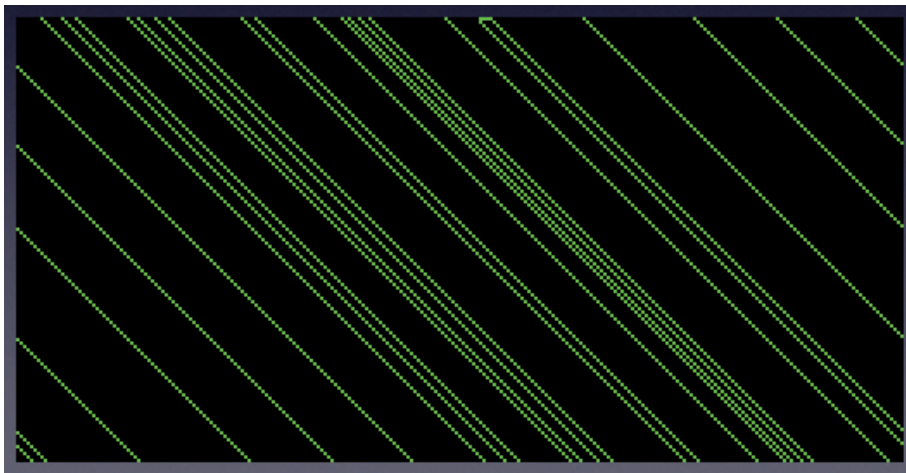


Figure 10: 56 号演化规则的进程

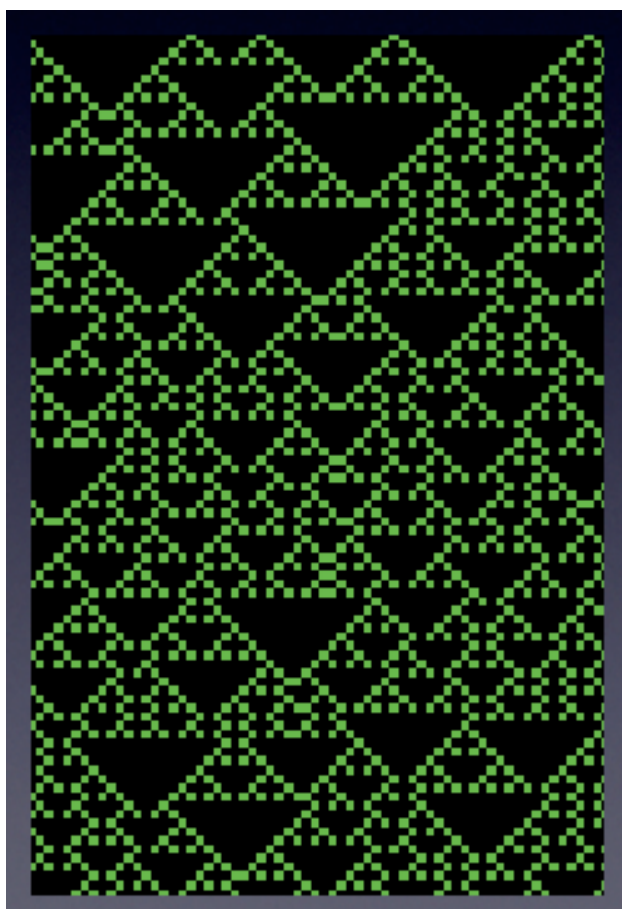


Figure 11: 18 号演化规则的进程

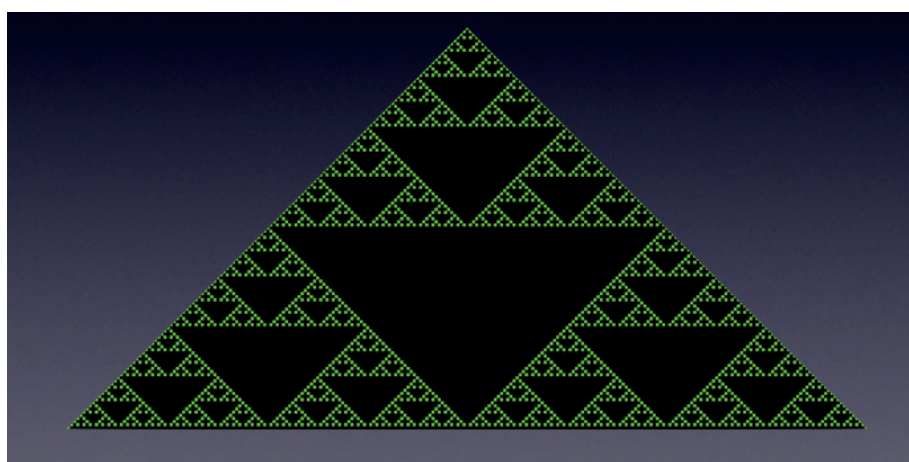


Figure 12: 18 号演化规则的进程

更复杂的是规则 110。我们连续看两个图,第一个是它在只有一个 1 的初始状态开始的,第二个是它在随机初始状态开始的,请看图13:

它体现出比混沌更复杂的规律。里面呈现出几大块区域,像打碎的玻璃。据说,110 号规则等价于一个通用图灵机。可以说,一切计算机所能做的计算工作,都可以通过 110 号规则配合适当的初始条件来做到。这个证明曾是保密的(现在不知公开了没有),而且据说和 Mathematica 里的一些保密算法有关。

Wolfram 对不同的规则,分成了四个种类。

- (1) 不动点。在这类规则下,元胞自动机在有限步的演化中,几乎全部的初始状态都会演化成稳定而不会改变的状态。
- (2) 周期环:在这类规则下,几乎全部初始状态都演化成周期性改变的图形。
- (3) 混沌:几乎全部的初始状态都演化成混沌的,非周期性的图形,而且往往具有自相似性。初始条件的微小变化往往导致若干步后的结果的显著不同。
- (4) 复杂:可以产生持续不断的复杂结构,难以描述和预测。

这个分类是经验的,并不能严格地证明。对这种一维的元胞自动机有许多推广的研究,譬如可以考察更广范围的邻居,或者依某种概率参数来随机地决定演化结果。但是几乎所有的演化结果,都可以分到这四类的某一类之中。

这是一个复杂一些的规则,考虑了距离为 2 的邻居。生成的结果也很有趣。

这个图演示了更复杂的规则,元胞可以处于三种状态(用不同颜色区分)。

对一些呈现出“复杂”状态的规则而言,甚至人们并不能确信,是否在足够多步以后,它们会变成别的状态,例如不动点?

Wolfram 写了一本书,叫做 *A New Kind of Science*。是专讲元胞自动机的各种应用。里面的研究包括海螺的花纹,包括交通流的性质,包括金融市场的“内在”随机性,等等。大家有兴趣可以去看,有专门的网站,可以看电子版。也可以下载得到。

在此我们还可以专门提一下,二维的元胞自动机,更可能在很简单的规则下,产生出奇怪的结果来。最简单的恐怕就是 Chris Langton 和 Greg Turk 提出的“蚂蚁”了。

顺便说一下,他是 SFI(Santa Fe Institute)的成员。SFI 是对研究复杂系统的很领先的研究机构,包括复杂系统的动力学,对社会和金融市场的模拟和研

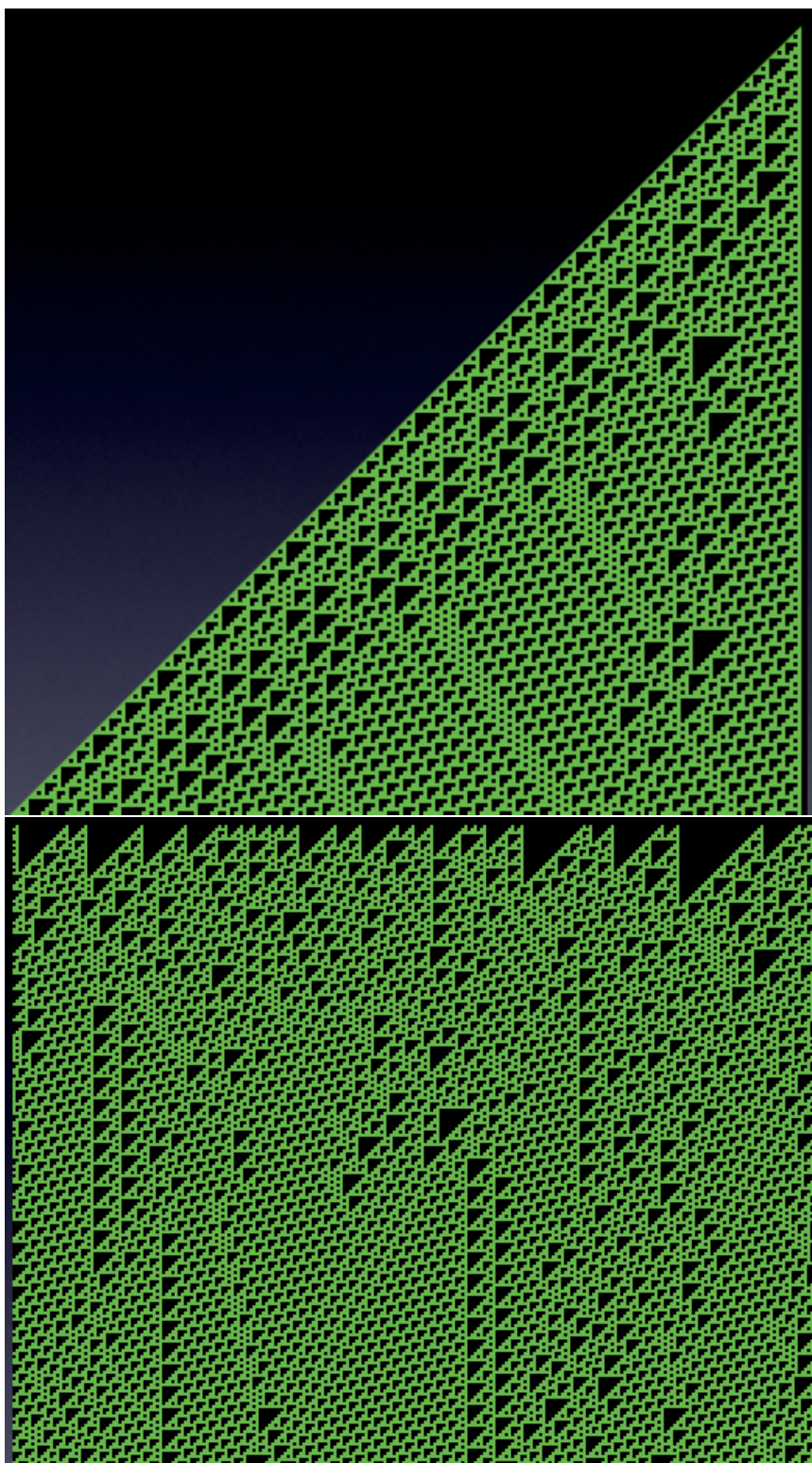


Figure 13: 110 号演化规则的进程



Figure 14: 更复杂的演化规则,考虑距离为 2 的邻居

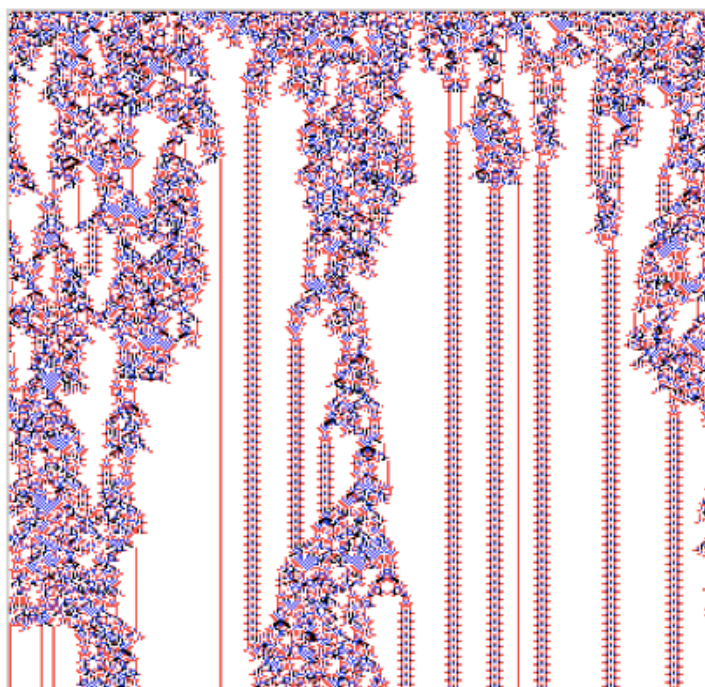


Figure 15: 更复杂的演化规则,考虑距离为 2 的邻居,元胞可以处于 3 种状态,表示为不同颜色

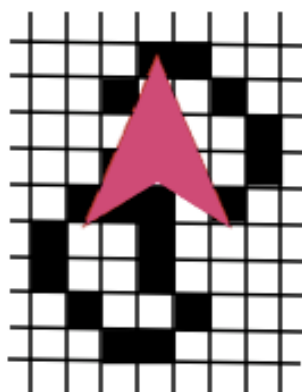


Figure 16: 100 步的蚂蚁



Figure 17: 400 步的蚂蚁

究、人工智能等问题都有很前沿的研究工作。官方网站是www.santafe.edu, 大家有兴趣的话可以去看。

假设一个“蚂蚁”在平面的方形网格中爬行,每个格子有两个状态:白色或黑色。蚂蚁爬进一个白色的格子时,就左转 90 度,并且把这个格子改成黑色。蚂蚁爬进一个黑色的格子时,就右转 90 度,并且把这个格子改成白色。最初时,整个“世界”都是白色的。

我们可以预测一下,这个蚂蚁会如何运动?

这是不到 100 步的时候,蚂蚁留下的图案:

这是接近 400 步的时候,蚂蚁留下的图案(没有画元胞的边界线):

在大概 500 步以内,它不时地回到原点,留下中心对称的图案。但随后开始出现奇怪的现象:

在从 500 步左右开始,直到接近 11000 步,它的图案完全失去了对称性,并且变得乱七八糟。



Figure 18: 4000 步的蚂蚁



Figure 19: 10000 步的蚂蚁

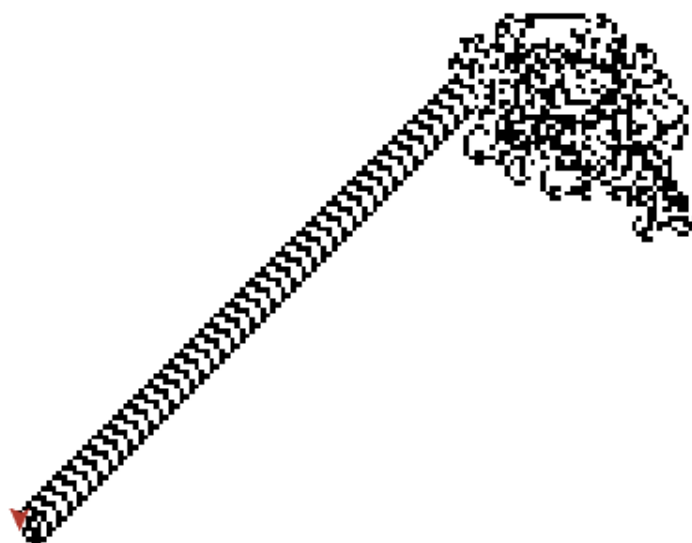


Figure 20: 蚂蚁的“公路”

在 11000 步左右,它的运动突然开始出现了规律。再往下:

每过 104 步,蚂蚁的位置向左下方推移一“节”。这种现象有时被叫做“公路”。

其实可以通过一个很简单而巧妙的分析,证明蚂蚁的轨道必然是无界的。这个证明大家可以试试。而且蚂蚁的轨道无论多么复杂,把时间倒转过来,都是完全可逆的。

在平面上如果存在两只或多只蚂蚁,情况会变得非常复杂。随着两只蚂蚁的初始位置不同,情况会不同,但多数情况下,蚂蚁都会通过建造“公路”的方式远离,但有时也会出现周期很大的周期轨道。也就是过几万步之后,居然能够完全回到初始状态。对此类问题的研究也有很多。

以上讲的是比较传统的元胞自动机例子。对它的研究虽然有许多,但总的来说,如果直接用它来对付实际问题,其思路不是很容易理解,也不是很容易推广。即使是 Wolfram 的那本 *A New Kind of Science*,也有许多批评的声音。但元胞自动机的计算机实验非常好做,有一个开源软件 Netlogo,也可以简单地做一些可视化的演示。我在上面的蚂蚁等图案都是用 Netlogo 生成的。

而下面我们要讲的,则是一种容易理解的元胞自动机。它来源于对实际问题的模拟,有时被称为格子气自动机, LGA (Lattice Gas Automata)。

LGA 是为了进行流体力学的计算而发展的离散方法。假设流体是由许多

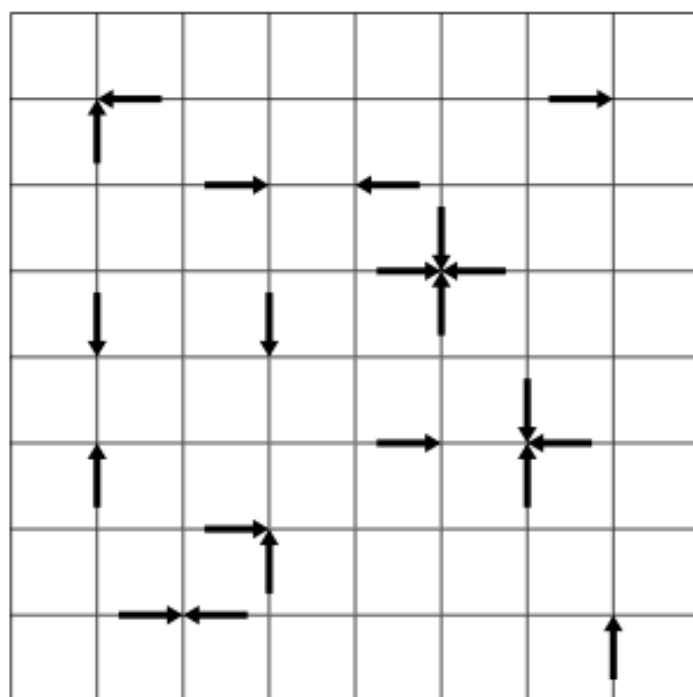


Figure 21: LGA

颗粒组成的,而不是连续的。再假设每个颗粒都在网格上运动,而不是在连续的空间中。而且时间也是离散的。如图21所示。

从这个角度上看,这个设定完全符合元胞自动机的定义。每个元胞就是空间的网格,状态有“被粒子占据”或“不被粒子占据”两种(其实更精确地说,粒子的运动方向也是要说到)。这里的关键是粒子的运动规则。在简单的问题中(譬如静态气体中的扩散等问题),甚至完全不需要考虑粒子的速度,只需要考虑运动方向就够了。

首先,其碰撞的规则必须是一种“变换”。如果所有粒子匀速运动并可以互相穿过,这就体现不出来流体的“流动”了。所以在粒子碰撞的时候,其运动方向应发生变化。但这同时要保持质量守恒和动量守恒。在正方形网格的假设下,最简单的规则就是 **HPP** 规则,见图22

如果考虑到容器壁,那就加上反射的规则。

HPP 规则已经能体现出和真实气流很类似的运动趋势了。譬如初始条件是绝大多数地方的粒子密度都是基本均匀的,而且运动方向都是随机设定的。而在中间有一团密度极大的粒子,我们可以明显地看到密度以波的形式在向四周扩散。波阵面碰到容器壁会反射。

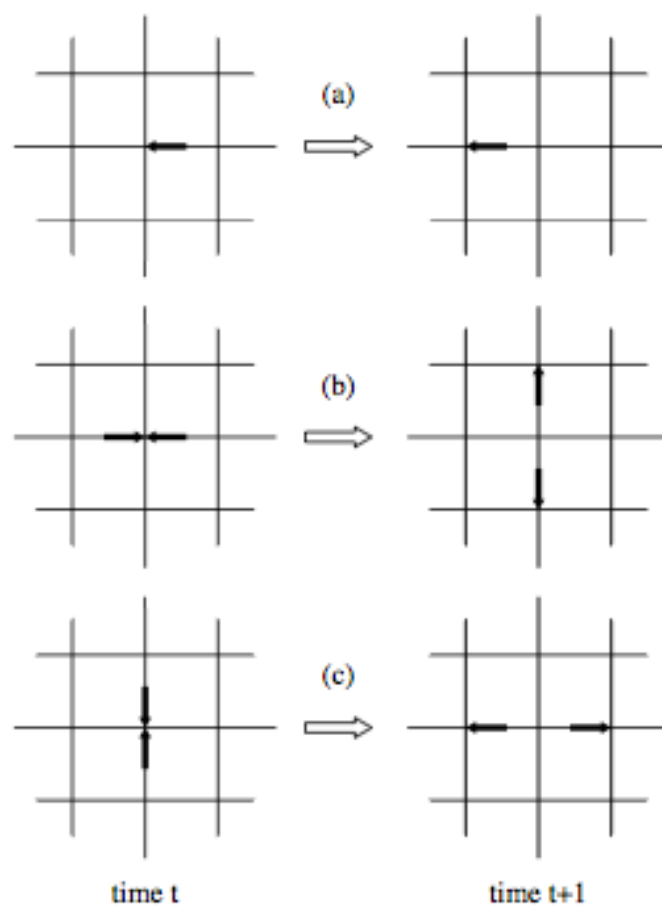


Figure 22: HPP 规则

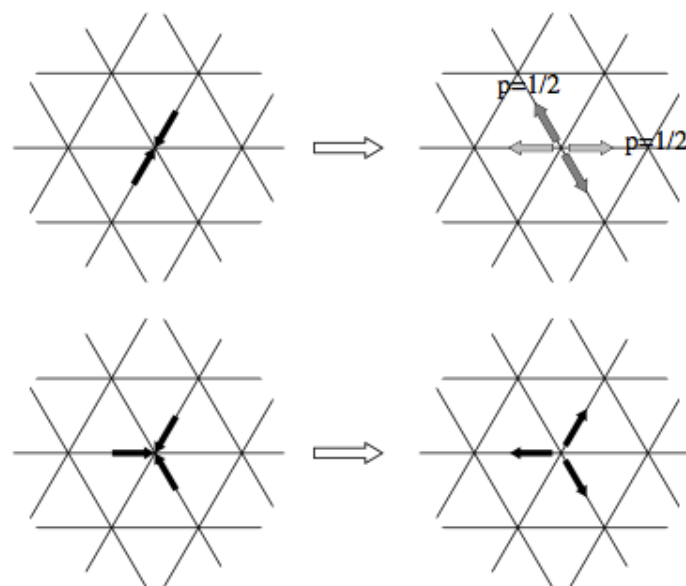


Figure 23: FHP 规则

但其在定量的角度上还不太好。主要是各个方向并不同性,在垂直方向和斜向的扩散性质不同。解决的方法是使用正三角形的网格,并配合稍复杂些的碰撞规则,这就是 FHP 规则。

FHP 规则其实有一些变种,这个图里讲的是,两个粒子碰撞,将随机生成其可能的结果之一。当然,无论什么规则的变种,质量守恒和动量守恒总是必须遵守的。

FHP 的计算结果更好,很好地模拟出了流体的运动状态,见图24

当然想投入工程应用,这里还有大量的问题。首先,这样设置碰撞规则,其合理性到底有多大? 光有动量守恒是不一定够的。当然 FHP 确实是合理的,对其分析涉及到连续化等数学处理,我们这次不多谈了。还有边界条件该怎么处理,等等问题。

LGA 在工程使用当中,还有一个弱点:噪声太大。每个元胞(也就是格点)不是被占据(1)就是不被占据(0),所以要设定大量在统计上等效的初始条件,进行反复的计算,最后求平均,才能较准确地算得不同点的流场情况。对此问题的解决方法有格子玻尔兹曼方法(Lattice Boltzmann Methods, LBM)等,它是一个有很大实用价值,尤其适用于复杂边界条件的计算流体力学方法。

LGA 的变种还包括 DLA 模型的模拟等。其思路非常简单:如果运动着的粒子带一定的粘性,在碰撞的时候会有一定概率粘结在一起,那么最终会粘

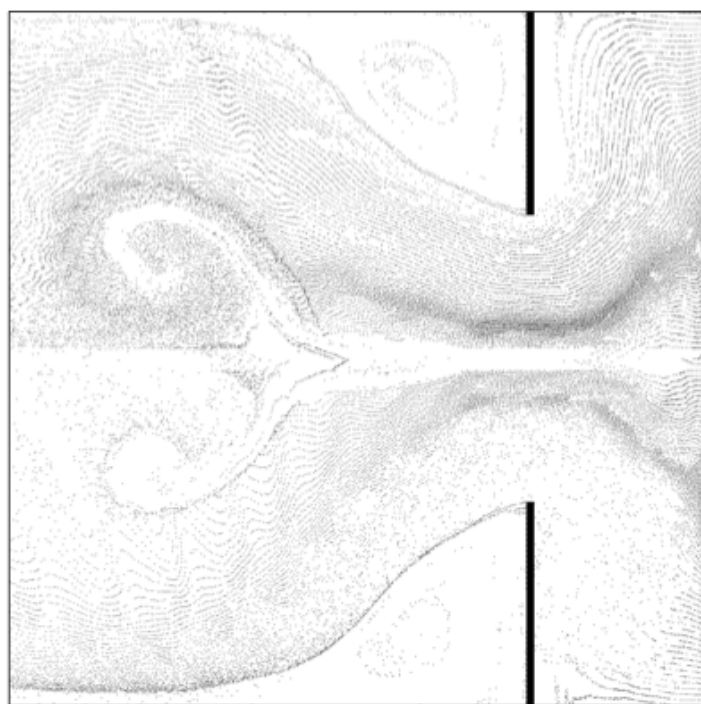


Figure 24: FHP 规则的计算结果



Figure 25: 岩层中的矿物结晶

结成一个什么样子？为了明确起见，我们可以在空间中央设定一个固定不动的“种子”，然后随机运动的粒子碰到种子就以一定概率粘结起来。这在一定程度上模拟了结晶的行为。

DLA 模型可以逼真地生成很像植物的结晶或者雪花的样子。图25是岩层中的矿物结晶。

图26:一行是在三角形网格上生成的 DLA 模拟结果，一行是真实的雪花照片。

LGA 的变种甚至还可以模拟出反应-扩散方程的解，或者多相流等许多复杂的问题。事实上，Ising 模型或 Potts 模型的模拟也可以归于元胞自动机里。还有著名的沙堆模型和渗流问题等。这些问题涉及到物理背景，讲起来比较复杂，我们有机会专门去谈。

实际上，在我们的建模工作中，可能最适宜于元胞自动机发挥作用的，要属交通流的问题了。

交通流是一种复杂的流动现象。首先它在一定程度上是离散的（无论车或人，一个一个都是分离的），而且每个个体的行为非常复杂，比如前面车的运动对后面的车有影响，但反过来几乎没有。车辆的加速减速也比无生命的流体要复杂得多。用连续的方程来描述，有一些障碍难以克服。

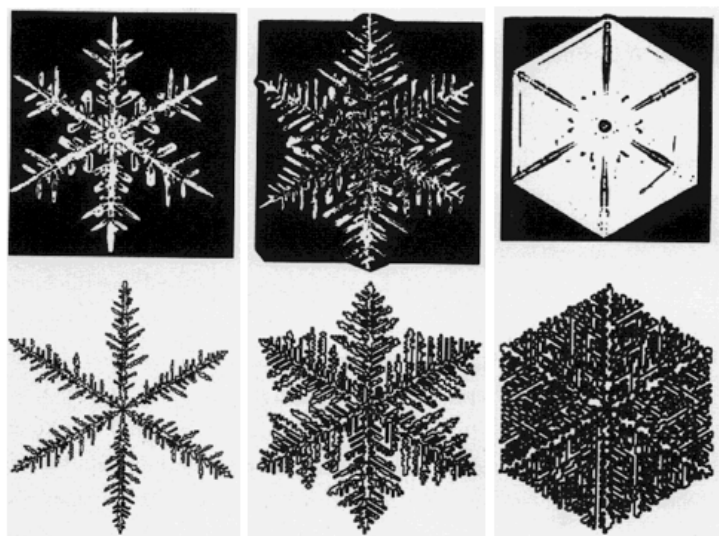


Figure 26: DLA 与雪花照片

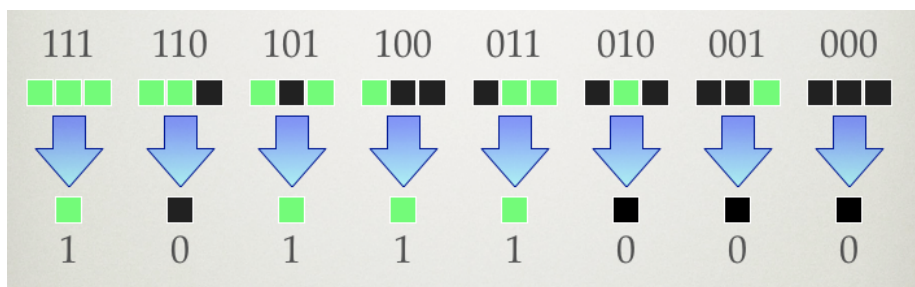


Figure 27: 规则 184

而元胞自动机,首先它是完全离散的。而且即便微观运动规律很复杂,我们充其量把元胞的运动规则搞得精致一些就足够了。所以元胞自动机非常适合于这类问题。

我们在最简单的假设下开始探讨问题:假设一维的道路由格子组成。每辆车占据一个格子。当时间从 t 变成 $t+1$,而且 t 时刻,车前的格子没有被占据,那么车向前走一个格子。

它的物理意义在于,在一维的道路上,车的视野受限,他不能越过前面的车看到更前面的情况。还有就是由于安全因素,所以只有当前面的位置已经空出来后,他才能够启动并且前进一格。

这个假设意味着什么呢? 在图示27里,我们用绿色表示车辆占据的格子,而黑色表示空出来的格子。

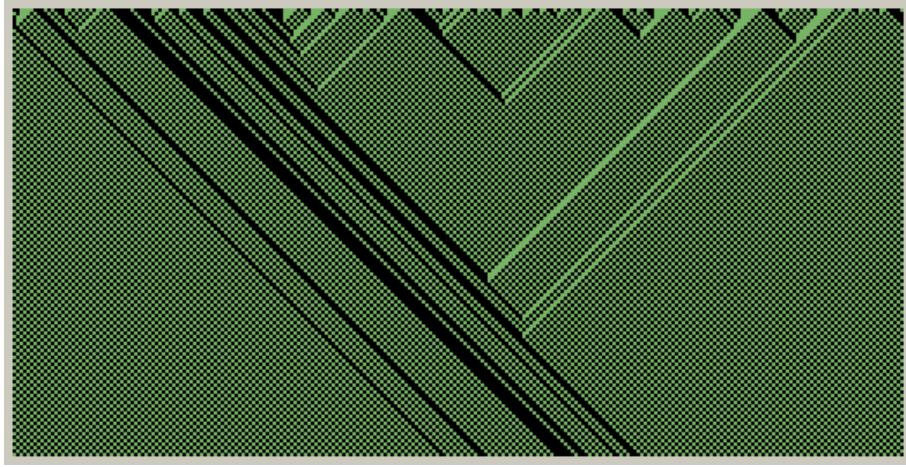


Figure 28: 规则 184 随时间的结果

这正是开头讲的 Wolfram 的 184 号规则。而其模拟结果,虽然不能直接应用于工程问题,但至少能够表现出交通堵塞的现象。请看图28,里面绿色条纹是车辆密度大的地方,我们可以看到堵塞的位置随时间在移动。当然这么简单的假设,还没有实用价值。有价值的是著名的 N-S 模型(Nagel - Schreckenberg model)。

为简单起见,我们暂时只研究一维的道路情况。比如高速公路,没有交叉路口。道路被分为等距的格子,每个格子或是空的,或被一辆车占据。车辆的速度可以取 $0, 1, 2, \dots, v_{max}$ 等整数值,其中 v_{max} 是最大速度。

建模的基本想法是:车辆总希望以尽量大的速度来行驶,但又不希望发生碰撞。据此可定出 N-S 模型的基本假设,也就是每辆车在时间 $t \rightarrow t+1$ 的时候的运动规则:

- (1) 前进:第 i 辆车的位置是 x_i ,它正以速度 v_i 前进。那么它在下一个时刻到达 $x_i + v_i$ 的位置: $x_i \rightarrow x_i + v_i$;
- (2) 加速:若第 i 辆车的速度 v_i 未达到最大速度,则其速度增加 1: $v_i \rightarrow \min(v_i + 1, v_{max})$;
- (3) 减速:若 v_i 超过其前方的空元胞数 d_i ,则 v_i 降低至 d_i ,以避免与前车碰撞: $v_i \rightarrow \min(v_i, d_i)$;
- (4) 随机慢化:运动车的速度 ($v_i > 1$) 以概率 p 降低 1,来模拟司机的过度反应且增加安全性: $v_i \rightarrow \max(v_i-1, 0)$ 。

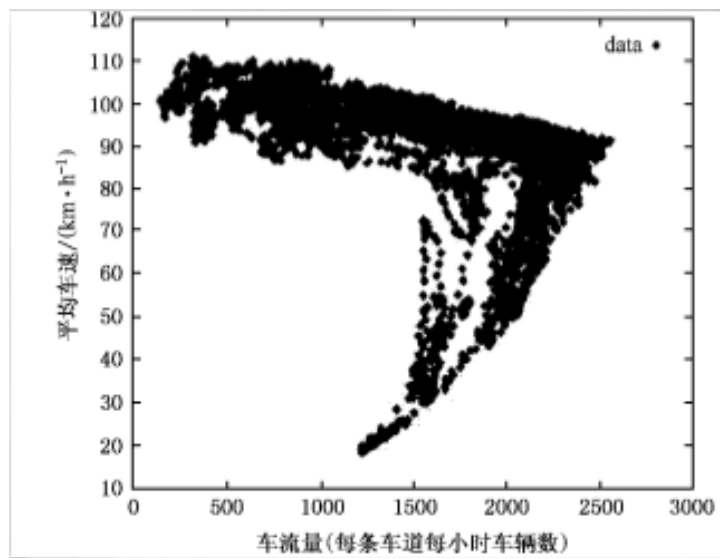


Figure 29: 车流量和车速的实测结果

只有这4条,其中第4条最为重要。司机不会任由车辆以可能的最高速行驶,所以会有一个减速的概率。

图29是我在一篇论文中拷贝下来的(忘记是哪篇了)。在美国加州一条公路上,对车流量和车速的实测结果的散点图。

而使用同样的速度限制等参数,用 N-S 模型模拟的结果如图30:

可以看到 N-S 模型的效果还是不错的。但是也有一些问题,比如达到道路最大流量时的车速,模型的结果和实测的结果还有不同。所以对 N-S 模型也有许多改进的方法。

最重要的改进都是集中在随机慢化的概率上。在 N-S 模型当中,随机慢化的概率是恒定的,但在实际问题中不然。一般来说,车速越快,司机做出随机慢化行为的概率越大。所以有的模型将随机慢化的概率做成车速的函数,这被称作 VDR 模型,其效用非常好。而且在实际操作当中,在车辆开始起步的阶段,随机慢化的概率显著超过已经匀速跑起来以后的阶段。所以也有一些模型是在这方面做文章。

即使是未经改进的 N-S 模型,实用效果已经比较明显了。由于其计算方便,在美国和德国,这个模型被用于对真实道路交通的研究和控制,比如控制信号灯,对交通堵塞的风险进行缓解等等。当然,对人流的情况、人车混行的情况等,其规律肯定会有显著的不同,同学们可以自己思考,也有许多资料可以搜到。

实际上,在近年来,美赛和全国赛的问题当中,有不少是可以使用元胞自

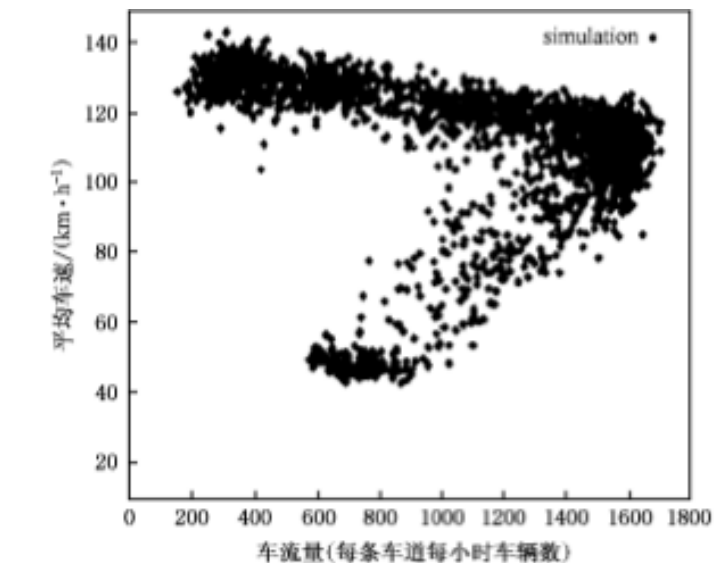


Figure 30: 车流量和车速的模拟结果

动机做模拟的。99 年美赛,逃生通路和房间容量问题;04 年全国赛,设计奥运场馆附近的交通情况;06 年美赛,登机问题;08 年美赛,交通环岛设计,等等。

在这些问题当中,核心的工作并不集中在模拟上。比如交通环岛的设计,那几乎就是一个优化问题,也就是考虑怎样设计交通标志最利于通行。但为了证实你的方法,增强说服力,进行一个计算机模拟,可能很有成效。有时甚至也是为了给自己开阔思路和验证模型,以实验的态度来做模拟,这也很有用处,只是要小心,如果对计算机编程不熟悉的话,在模拟上会花费太多时间,总时间很可能不够。

而元胞自动机,总的来说是一种较可靠的,便于思考的做模拟的框架。希望同学们能对此有所了解。

对元胞自动机的深入探讨,可以看如下书籍:Wolfram 的《A New Kind of Science》,电子版可看<http://www.wolframscience.com/nksonline/toc.html>。还有翻译出版的一本著作《物理系统的元胞自动机模拟》,Bastien Chopard 等著,祝玉学等译,清华大学出版社。里面详细地讲了 LGA 的种种应用和分析方法。

Netlogo 等软件可以帮助人们直观地了解元胞自动机的一些应用,但在真正的建模应用当中,还是自己编程要更可行一些。

我们这次就到这里吧,谢谢大家。