



北京大学

数学模型 Term Project

题 目： 行人疏散的社会力模型和

元胞自动机模型的初步比较分析

小组成员： 付博铭、李擎宇、张佳琦

学 号： 1500060606、1600060601、

1600060613

院 系： 数学科学学院

联系方式： bomingfu@pku.edu.cn

1600060601@pku.edu.cn

1600060613@pku.edu.cn

二〇一八年六月

关键词：行人疏散 社会力模型 元胞自动机模型 仿真模拟

摘要：

紧急情况下行人疏散问题是我们小组的选题。我们小组成员在开始研究之前，对该问题均不了解，通过文献收集，小组讨论，对行人疏散问题的社会力模型和元胞自动机模型有了较充分的认识，并使用MATLAB运行了这两个模型。我们从模型假设、模型构成和模型应用介绍了两个模型的不同，对两个模型各自的不足和改进做了讨论。

目录

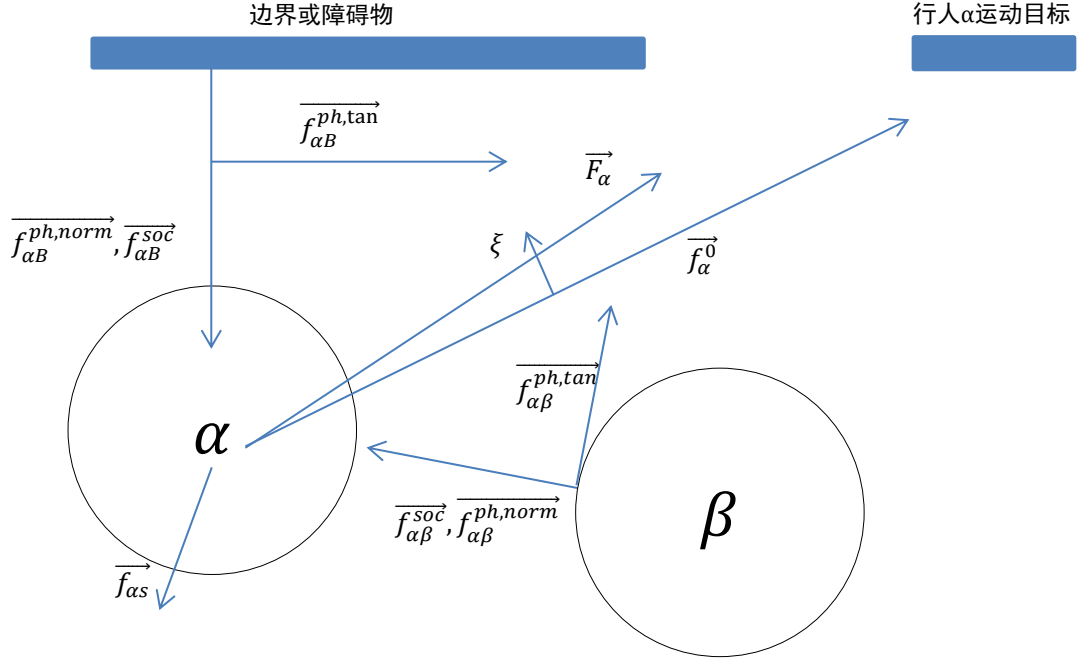
1.社会力模型.....	3
1.1 社会力模型的基本原理.....	3
1.1.1 自驱动力.....	3
1.1.2 排斥力.....	4
1.1.3 其他力.....	7
1.2 社会力模型的参数选取.....	7
1.2.1 自驱动力参数.....	7
1.2.2 行人之间作用力参数.....	8
1.2.3 行人与障碍物作用力参数.....	9
1.3 社会力模型MATLAB仿真	9
1.4 社会力模型的优势、不足和改进.....	12
2.元胞自动机模型.....	13
2.1 元胞自动机简介.....	13
2.2 元胞自动机模型的建立.....	13
2.2.1 静态场.....	14
2.2.2 动态场.....	15
2.2.3 排斥、摩擦和惯性行为.....	16
2.2.4 行人速度.....	18
2.3 元胞自动机模型参数选取和MATLAB仿真	18
2.4 元胞自动机模型的优势、不足和改进.....	20
参考文献.....	21
附录A.....	23
附录B.....	27

模拟行人在一个空间的疏散是在二维平面上进行的，现有模型基本分为连续性模型和离散性模型。连续性模型的一个典型代表是社会力模型，该模型对行人进行受力分析，核心是将人的社会行为所产生的一种心理作用表示为力，来建立牛顿力学的运动方程。离散性模型的一个典型代表是元胞自动机模型，考虑行人的运动规则，核心是引入“场”的概念确定向邻域元胞的转移概率。

1. 社会力模型

1.1 社会力模型的基本原理

对于行人 α 做受力分析，如图所示：



考虑行人受到的所有物理力和社会力，建立牛顿力学方程如下：

$$m_\alpha \frac{d\vec{v}_\alpha}{dt} = \vec{F}_\alpha(t) + \xi$$

$$\vec{F}_\alpha(t) = \vec{f}_\alpha^0(\vec{v}_\alpha, v_\alpha^0 \vec{e}_\alpha) + \sum_\beta \vec{f}_{\alpha\beta}(\vec{e}_\alpha, \vec{r}_\alpha - \vec{r}_\beta) + \sum_B \vec{f}_{\alpha B}(\vec{e}_\alpha, \vec{r}_\alpha - \vec{r}_B) + \sum_s \vec{f}_{\alpha s}(\vec{e}_\alpha, \vec{r}_\alpha - \vec{r}_s, t)$$

其中： \vec{f}_α^0 为自驱动力， $\sum_\beta \vec{f}_{\alpha\beta}$ 为行人与其他行人作用的合力， $\sum_B \vec{f}_{\alpha B}$ 为行人与边界或障碍物的作用力合力， $\sum_s \vec{f}_{\alpha s}$ 为某种吸引力的合力。 m_α 为行人 α 的质量， \vec{v}_α 为当前实际速度， v_α^0 为行人 α 的期望速度大小， \vec{e}_α 为行人 α 的期望方向， \vec{r}_α 和 \vec{r}_β 为行人 α 和 β 的位置矢量， \vec{r}_B 和 \vec{r}_s 为边界和吸引点的位置矢量， ξ 为随机变量，在模型中代表不确定的行人行为， t 为时间。

由此，我们可以确定 t 时刻的行人速度和加速度，更新为 $t + \Delta t$ 时刻，便可以确定行人运动轨迹，就可以顺利模拟行人运动。

1.1.1 自驱动力

自驱动力是模型中最显著的力，它决定行人是否能以期望速度向目的地运动。没有干扰

下，行人 α 将以期望速度 $v_\alpha^0 \vec{e}_\alpha$ 向目标运动，不会选择绕行。值得注意的是，行人 α 的目标通常是一扇门或一个区域而非一个点，此时的目标点假定为门或区域离行人 α 最近的点，在实际情形中，行人的目标通常不止一个。下式为 t 时刻行人 α 的自驱动力方向 $\vec{e}_\alpha(t)$ 的表达式：

$$\vec{e}_\alpha(t) = \frac{\vec{r}_\alpha^k - \vec{r}_\alpha(t)}{\|\vec{r}_\alpha^k - \vec{r}_\alpha(t)\|}$$

其中行人 α 的第 k 个目标点为 \vec{r}_α^k ， $\vec{r}_\alpha(t)$ 为其 t 时刻位置点。

考虑到实际情况中存在必要的减速和躲避行为，行人 α 的实际速度 $\vec{v}_\alpha(t)$ 与期望速度 $v_\alpha^0 \vec{e}_\alpha$ 的差通过一定的“松弛时间” τ_α 来修正，以接近期望速度，用加速度形式表示如下：

$$\vec{f}_\alpha^0(\vec{v}_\alpha, v_\alpha^0 \vec{e}_\alpha) = m_\alpha \frac{v_\alpha^0 \vec{e}_\alpha - \vec{v}_\alpha}{\tau_\alpha}$$

其中 \vec{v}_α 为实际速度， v_α^0 为期望速率， τ_α 为松弛时间，反应行人加减速的能力。

以上行人期望方向和自驱动力表达式均假定行人是独立的，不受其他行人影响。然而在实际运动过程中，行人往往会出现从众行为，跟随周围行人运动。为此引入方向权重函数来描述行人 α 的从众程度：

$$\vec{e}_\alpha(t) = \text{Norm}[(1 - n_\alpha) \vec{e}_\alpha^* + n_\alpha \langle \vec{e}_\beta(t) \rangle_\alpha]$$

其中 n_α 为从众系数，取值范围[0,1]，其值越大，表示行人 α 越从众，为0时表示行人 α 自由行进； \vec{e}_α^* 为个体方向向量； $\langle \vec{e}_\beta(t) \rangle_\alpha$ 为周围人行进的平均方向； $\text{Norm}(z) = z/\|z\|$ 。

1.1.2 排斥力

排斥力包括两部分，分别来自行人之间的相互作用和行人与墙壁等障碍物的相互作用。

(1) 行人之间排斥力

在人群中，行人 α 通常会与其他人保持一定的距离，人类学家霍尔称之为“空间关系学”。当行人 α 运动需求空间内存在其他行人时，行人 α 通常会感到不舒适，其他行人对行人 α 便产生斥力效应，这种斥力效应主要包括心理排斥力和物理作用力两部分。考虑 β 对 α 的排斥力，即：

$$\vec{f}_{\alpha\beta}(t) = \vec{f}_{\alpha\beta}^{soc}(t) + \vec{f}_{\alpha\beta}^{ph}(t)$$

其中社会力 $\vec{f}_{\alpha\beta}^{soc}(t)$ ：

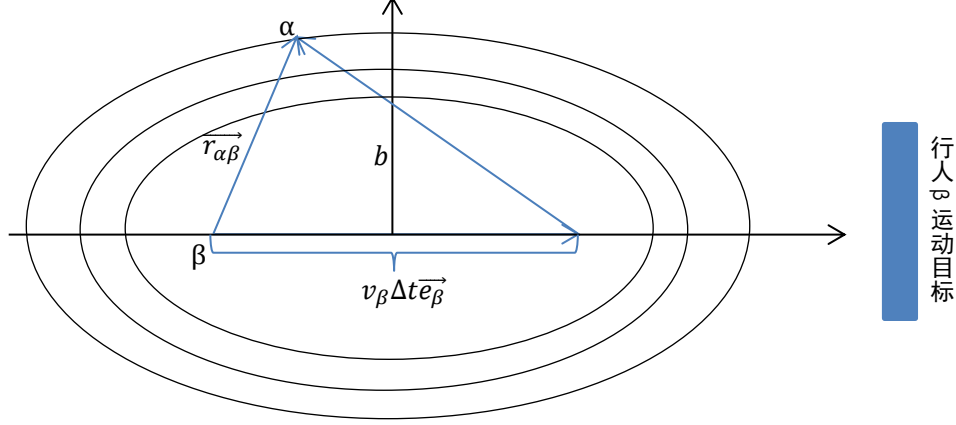
$$\vec{f}_{\alpha\beta}^{soc}(t) = -\nabla_{\alpha\beta} V_{\alpha\beta}(b(\vec{r}_{\alpha\beta}))$$

该模型假设行人 α 与其他行人间潜在斥力 $V_{\alpha\beta}(b)$ 随着变量 $b(\vec{r}_{\alpha\beta})$ 的减小，呈现单调递减

的趋势。考虑到行人下一时间的移动需要一定空间这一实际情况，模型设定一个椭圆形等势线，其方向指向目标点，该椭圆的短半轴计算公式如下：

$$2b = \sqrt{(\|\vec{r}_{\alpha\beta}\| + \|\vec{r}_{\alpha\beta} - v_{\beta}\Delta t\vec{e}_{\beta}\|)^2 - (v_{\beta}\Delta t)^2}$$

$\vec{r}_{\alpha\beta} = \vec{r}_{\alpha} - \vec{r}_{\beta}$ 是行人的位置矢量差； $v_{\beta}\Delta t$ 是行人 β 在 Δt 时间内的步长。



上图表示出了椭圆短半轴 b 。尽管如此，上述公式仍然不够直观，无从下手，因此我们对上述社会力公式进行如下的简化，用指数函数来表示，用相互作用强度和作用范围来修正：

$$\vec{f}_{\alpha\beta}^{soc}(t) = A_{\alpha} \exp\left(\frac{r_{\alpha\beta} - d_{\alpha\beta}}{B_{\alpha}}\right) \vec{n}_{\alpha\beta}$$

其中， A_{α} 和 B_{α} 是常数，表示行人 α 与其他行人的相互作用强度和作用范围； $r_{\alpha\beta}$ 是相互作用两人的半径和； $d_{\alpha\beta}$ 是行人 α 和 β 的间距； $\vec{n}_{\alpha\beta} = (n_{\alpha\beta}^x, n_{\alpha\beta}^y) = \frac{\vec{r}_{\alpha\beta}}{d_{\alpha\beta}}$ 是由行人 β 指向 α 的单位向量， $\vec{n}_{\alpha\beta}$ 和 $d_{\alpha\beta}$ 随时间变化。实际情况中，行人 α 更容易注意到前方行人的情况，而忽略后面行人的情况，因此前面行人对行人 α 的社会力要比后面行人对行人 α 的社会力大。为了表达这个影响，引入一个常数 λ_{α} ， $\lambda_{\alpha} \in [0,1]$ ，改进方向权重的表达式为：

$$w(\vec{e}_{\alpha}, \vec{f}_{\alpha\beta}^{soc}) = \lambda_{\alpha} + (1 - \lambda_{\alpha}) \frac{(1 + \cos \varphi_{\alpha\beta})}{2}$$

其中， $\varphi_{\alpha\beta}$ 为行人间社会力和期望方向夹角。因此，修正后的社会力为：

$$\vec{f}_{\alpha\beta}^{soc}(t) = A_{\alpha} \exp\left(\frac{r_{\alpha\beta} - d_{\alpha\beta}}{B_{\alpha}}\right) \vec{n}_{\alpha\beta} (\lambda_{\alpha} + (1 - \lambda_{\alpha}) \frac{(1 + \cos \varphi_{\alpha\beta})}{2})$$

接下来考虑物理力 $\vec{f}_{\alpha\beta}^{ph}(t)$ ：

物理力由行人身体相互接触而产生的作用力，当相互作用行人的半径和大于彼此之间距离时产生，即 $r_{\alpha\beta} > d_{\alpha\beta}$ 。物理力包括正压力 $\vec{f}_{\alpha\beta}^{ph,norm}(t)$ 和滑动摩擦力 $\vec{f}_{\alpha\beta}^{ph,tan}(t)$ 构成。正压力反抗其他行人的压缩力，避免身体被破坏，它是一个法向力。正压力的大小由两个行人之

间的距离和两行人的半径和所决定，正压力的大小与两个行人之间的距离成线性关系，距离越近，推力越大。滑动摩擦力来自一种趋向，是一种切向力，使行人在短距离内离开接触。这与物理中滑动摩擦力是不一样的，物理中滑动摩擦力是与正压力有关且与相对速度无关的量，但在这里切向力是行人 β 对行人 α 的带动的一种描述。滑动摩擦力的大小不仅与两个行人间的距离和两行人的半径和有关，还与两行人的切向相对速度有关，两行人之间的距离越近，滑动摩擦力越大；两行人间的速度差越大，滑动摩擦力就越大。

首先定义分段函数 $\theta(x) = \begin{cases} x, & x > 0 \\ 0, & \text{其他} \end{cases}$ ，行人距离小于半径和时， $\theta(x)$ 取 x ；其他情况下 $\theta(x)$ 取0，表示没有接触。 $\vec{t}_{\alpha\beta} = (-n_{\alpha\beta}^2, n_{\alpha\beta}^1)$ 表示切向，切向相对速度为 $\Delta v_{\alpha\beta}^t = (\vec{v}_\alpha - \vec{v}_\beta) \cdot \vec{t}_{\alpha\beta}$ ，它表示两个行人间的速度差在 $\vec{t}_{\alpha\beta}$ 上的投影大小，因此我们有：

$$\begin{aligned}\vec{f}_{\alpha\beta}^{ph,norm}(t) &= k\theta(r_{\alpha\beta} - d_{\alpha\beta})\vec{n}_{\alpha\beta} \\ \vec{f}_{\alpha\beta}^{ph,tan}(t) &= K\theta(r_{\alpha\beta} - d_{\alpha\beta})\Delta v_{\alpha\beta}^t \vec{t}_{\alpha\beta}\end{aligned}$$

其中 k 为正压力系数， K 为滑动摩擦力系数。

综上，法向上的正压力和切向上的滑动摩擦力共同组成了物理力，其表达式为：

$$\vec{f}_{\alpha\beta}^{ph}(t) = k\theta(r_{\alpha\beta} - d_{\alpha\beta})\vec{n}_{\alpha\beta} + K\theta(r_{\alpha\beta} - d_{\alpha\beta})\Delta v_{\alpha\beta}^t \vec{t}_{\alpha\beta}$$

(2) 行人与障碍物作用力

若一定范围内有障碍物如墙体、栏杆、其他建筑物等存在，行人有与其保持一定距离的心理作用。在运动过程中，行人越接近边界会觉得越不舒适，因为行人此时会把注意力更多地放在避免受伤害方面。此时边界引起的斥力效应，用行人与障碍物的排斥力表示，主要包括与障碍物的心理排斥力和物理作用力两部分。只有当行人与障碍物之间的距离小于行人的半径时，即 $r_{\alpha\beta} > d_{\alpha\beta}$ 时才产生物理力。

在社会力模型中，行人与障碍物之间的作用力与行人之间的排斥力表达方式相似，由正压力和滑动摩擦力构成。但是行人在遇到其他行人和遇到障碍物之人后反应不同，这需要通过选择不同的参数来使仿真更加贴合实际。

因此我们有，行人 α 与障碍物 B 之间的作用力表达式为：

$$\vec{f}_{\alpha B}(t) = \vec{f}_{\alpha B}^{soc}(t) + \vec{f}_{\alpha B}^{ph}(t)$$

$\vec{f}_{\alpha B}^{soc}(t)$ 是心理排斥力， $\vec{f}_{\alpha B}^{ph}(t)$ 是物理排斥力，仿照上面讨论我们有：

$$\vec{f}_{\alpha B}^{soc}(t) = A_B \exp\left(\frac{r_{\alpha B} - d_{\alpha B}}{C_B}\right) \vec{n}_{\alpha B}$$

$$\overrightarrow{f_{\alpha B}^{ph}}(t) = k\theta(r_{\alpha B} - d_{\alpha B})\overrightarrow{n_{\alpha B}} + K\theta(r_{\alpha B} - d_{\alpha B})\langle \overrightarrow{v_{\alpha}}, \overrightarrow{t_{\alpha B}} \rangle \overrightarrow{t_{\alpha B}}$$

式中， A_B 是行人与障碍物相互作用力强度； C_B 是行人与障碍物相互作用范围； $d_{\alpha B}$ 是行人与障碍物之间的距离； $\overrightarrow{n_{\alpha B}}$ 是单位法向量，方向由障碍物指向行人 α ， $\overrightarrow{t_{\alpha B}}$ 是单位切向量，方向与障碍物平行； $\langle \overrightarrow{v_{\alpha}}, \overrightarrow{t_{\alpha B}} \rangle$ 是行人速度在障碍物方向上的投影。

1.1.3 其他力

(1) 吸引作用力

行人与行人，行人与其他物体之间存在吸引作用。地点 \vec{r}_s 处的吸引力 $\overrightarrow{f_{\alpha s}}$ 可用与排斥力相似的单调递增函数 $W_{\alpha s}(\|\overrightarrow{r_{\alpha s}}\|, t)$ 表示： $\overrightarrow{f_{\alpha s}^{attach}}(\|\overrightarrow{r_{\alpha s}}\|, t) = -\nabla W_{\alpha s}(\|\overrightarrow{r_{\alpha s}}\|, t)$ ，式中 $\overrightarrow{r_{\alpha s}} = \overrightarrow{r_{\alpha}} - \overrightarrow{r_s}$ 为行人 α 与吸引点位置矢量差。行人之间吸引力可用常数 $E_{\alpha\beta}$ 与向量 $n_{\alpha\beta}$ 的积表示，表达式为 $\overrightarrow{f_{\alpha\beta}^{attach}} = -E_{\alpha\beta}n_{\alpha\beta}$ 。总的吸引力为前两式的和。

吸引作用在模型中可以视为排斥力的一部分（反方向的一部分），稍有不同的是吸引力的作用范围比行人间作用力要大，且吸引力是不稳定的，可能会衰减。人群相互作用力与分子间作用力相似，考虑吸引作用更符合人群的实际情况，但这并不是必须的，在紧急情况下，即使忽略这一部分力是完全可以的。

(2) 随机变量 ξ

行人的行为具有随机性并受到随机影响，比如左右都可通过时的随机选择以及踩踏摔倒等突发情况的产生。假设在一般情况下，则随机项的影响完全可以忽略不计，并不影响模拟。

因此，对于以上两项，我们在MATLAB运行时均选择了忽略。

1.2 社会力模型的参数选取

模型参数的选取直接影响仿真效果，我们综合参考文献选定参数。

1.2.1 自驱动力参数

(1) 行人尺寸和质量 m_{α}

假设行人的形状近似为圆形，半径取值则主要参考人体的肩宽和胸厚尺寸。行人的肩宽大于胸厚，男性肩宽和胸厚均大于女性。若将行人模拟为圆形，以肩宽作为直径，本文推荐

行人直径尺寸在 $[40cm, 60cm]$ 符合正态分布, 质量在 $[50kg, 80kg]$ 符合正态分布。

(2) 松弛时间 τ_α

松弛时间与行人的反应时间以及惯性有关。人从接受刺激、认识到刺激再到快速做出反应, 所需要的时间非常短暂。一般条件下, 人对外界变化的反应时间约为 $0.1\sim 0.5s$, 对于复杂变化反应时间约 $1\sim 3s$, 要进行复杂判断和认识的反应时间约为 $3\sim 5s$ 。行人在移动过程中, 从实际速度向期望速度的加速过程并不复杂, 将松弛时间取值为 $0.5s$ 即可。

(3) 期望速度 v_α^0

不同个体期望速度不同, 可以描述为符合某种分布的随机变量。但是Weidmann(1992)将期望速度表示成行人密度的函数, 这个做法是符合实际的, 正常状况下行人平均速度与密度的关系式如下: $v(\rho) = v_0\{1 - \exp[-1.913m^2((1/\rho) - (1/\rho_{max}))]\}$, 其中 v_0 为期望速度, $v(\rho)$ 是行人运动的平均速度, ρ 为测得的行人密度, 单位 $person/m^2$, ρ_{max} 取值为 $5.4person/m^2$ 。不同疏散场景下行人期望速度不同, 因此可以采用统计的方法标定一些场景的行人期望速度。

(4) 从众系数 n_α

从众系数 n_α , 取值范围 $[0,1]$, 可以用正态分布来刻画行人从众程度。

1.2.2 行人之间作用力参数

(1) 行人之间的作用强度 A_α 和作用范围 B_α

在计算行人间相互作用力时, 行人之间的作用强度 A_α 和作用范围 B_α 有两种取值方法。一种是考虑行人质量, 另一种是将行人质量视为单位质量时, 用加速度来描述, 单位为 $m \cdot s^{-2}$, 相关文献的取值有 $2.1m \cdot s^{-2}$ 和 $5m \cdot s^{-2}$ 。我们在模拟时, 做了进一步改进, 取 $A_\alpha = m_\alpha \times 2.1m \cdot s^{-2}$ 。根据相关文献, 作用范围的取值有 $B_\alpha = 0.3m, 0.08m, 0.1m, 0.5m$, 这里我们取 $B_\alpha = 0.08m$ 。

(2) 正压力系数 k 和滑动摩擦力系数 K

行人之间接触时, 要考虑正压力系数 k 和滑动摩擦力系数 K , Helbing(2002)中的社会力模型没有限定的压缩范围, 令 $k = 1.2 \times 10^5 N \cdot m^{-1}$, 用这个很大的值保证了行人在拥挤时不会出现重叠, 然而Lakoba(2005)指出这个 k 值不合实际, 如果按此取值, 两个行人彼此挤压 $5cm$ 就要承受 $6000N$ 的压力, 这个力足以将行人摧毁或者击飞。因此, 模拟行人运动的挤压过程, 需要设定一个行人的身体能够承受的最大压缩量, Lakoba将这个量设为身体尺寸

的20%，当压缩量达到人的最大承受能力时，除以压缩量便是正压力系数。根据文献，正常人体的肩宽乘以20%，取 $0.1m$ ，此时达到的最大压力为 $4000N$ ，则 $k = 4 \times 10^4 N \cdot m^{-1}$ ，相应的滑动摩擦力系数 $K = 6 \times 10^4 N \cdot m^{-1}$ 。

1.2.3 行人与障碍物作用力参数

在社会力模型中，行人与障碍物之间的作用力与行人之间的排斥力表达方式相似，但是行人在遇到其他行人和遇到障碍物之人后反应不同，要改变参数 A_B 和 C_B 的选取。

考虑行人与障碍物之间的作用力强度 A_B ，不同于行人之间作用力，行人对障碍物不会产生反作用力。通常，行人在行走时通常会远离障碍物，这个距离要比与其他行人间的距离大得多，因此 A_B 取值要比 A_α 大，*Helbing*(1995)对 A_α 的取值为 $2.1m \cdot s^{-2}$ ，对 A_B 的取值为 $10m \cdot s^{-2}$ 。实际情况下， A_B 与行人密度 ρ 有关， ρ 越大， A_B 越小，综合考虑，模拟时我们取 $A_B = m_\alpha \times 10 m \cdot s^{-2}$ 。

行人与障碍物相互作用范围 C_B 是行人移动时，潜意识地与障碍物或边界保持的一定距离，这个距离称为行人回避距离。 C_B 的取值显然与行人状态有关，应该在一个范围内取值，行人密度大时，行人会被迫接受贴近障碍物移动， C_B 此时较小，正常情况下， C_B 取值要大一些。*Daamen.W*(2002)对不同障碍物和边界的行人回避距离做了总结， $C_B \in [10cm, 45cm]$ ，据此，我们考虑一般情况，取 $C_B = 0.3m$ 。

1.3 社会力模型MATLAB仿真

至此，参数已经全部选定完毕。基于以上模型和参数，我们设定了简单情形下的行人疏散，考虑以下问题：不同期望速度下，行人的疏散时间。通过此问题来观察社会力模型的应用情况。单出口长方形房间的尺寸为 $8m \times 12m$ ，初始 50 个行人在房间内随机分布，出口为半径 $0.8m$ 的半圆周。

*Figure 1*是初始状态行人分布，图中蓝色线表示行人瞬时速度，绿色半圆周为出口，红色线表示边界，可以看到初始状态行人的速度方向均指向出口，当行人的圆心落入绿色半圆周内时，则将行人从空间中移出。

*Figure 2*是行人形成的聚集状态，图中可以看出，部分行人存在挤压等接触作用(如15和4号行人)，因此对行人之间相互作用力的刻画是符合实际的。

下面我们考虑了不同期望疏散速度下的全部疏散时间的变化情况，我们取定期望速度为

$v_{\alpha}^0 = 0.5、1.0、1.5、2.0、2.5、3.0 m/s$, *Figure 3*为不同期望速度下, 行人全部疏散时间 T 的 $v_{\alpha}^0 - T$ 图像, 可以明显看出, 图像存在极小值, 这意味着行人期望速度并不是越大越好, 过大的期望速度反而减小了全部疏散时间, 这也是与实际情形符合的, 因为这种情况下, 行人容易在出口处形成拥堵, 从而延长了疏散时间。

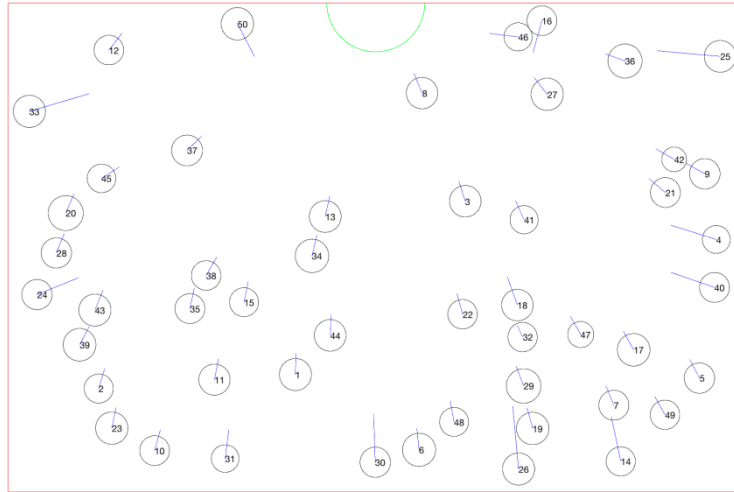


Figure 1 初始行人分布

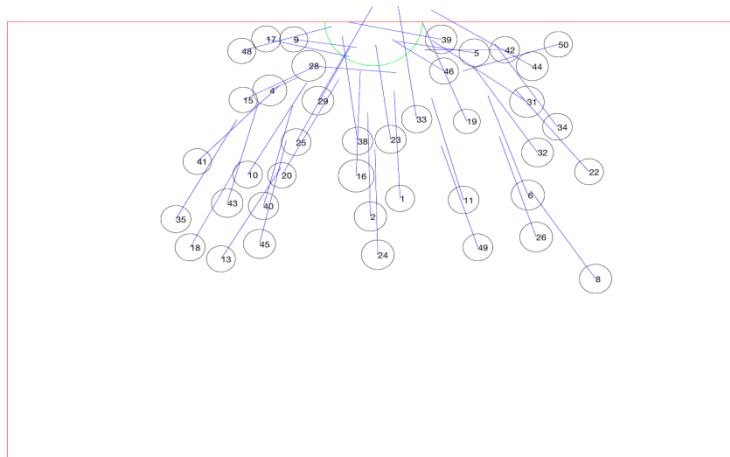


Figure 2 行人聚集情况

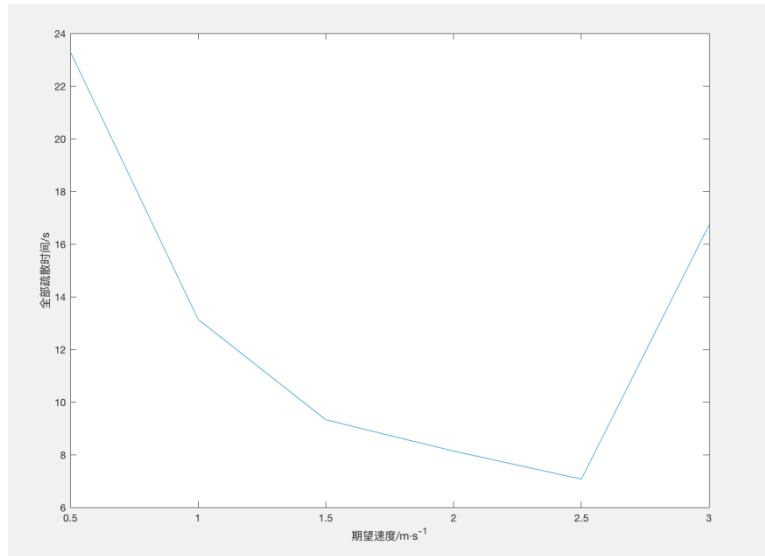


Figure 3 期望速度与全部疏散时间

接下来我们考虑了不同出口直径下，全部疏散时间的变化。分别设置了 $diam = 1.2\text{m}, 1.4\text{m}, 1.6\text{m}, 1.8\text{m}, 2.0\text{m}$ ，从Figure 4可以看出，疏散时间随着出口直径的增大而减小，斜率会发生变化，出口直径足够大时，行人疏散时间的变化会很小，此时边界的存在已经无法限制行人的自由移动，行人间作用力和行人与障碍物和边界作用力的影响无法明显改变行人自驱力的方向。

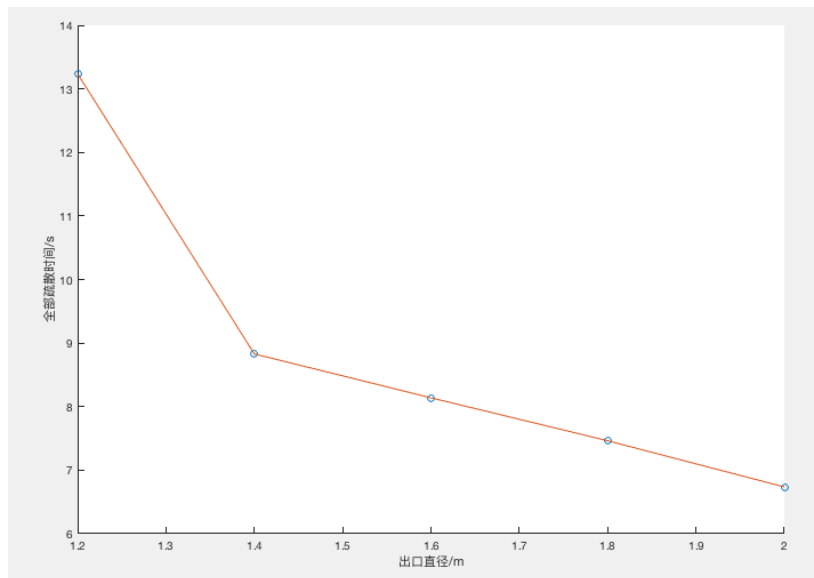


Figure 4 出口直径与全部疏散时间

1.4 社会力模型的优势、不足和改进

社会力模型用牛顿力学方程描述行人在复杂环境下的运动过程,明确了行人运动的产生源于自身的主观行动力(包括自驱动力和心理力)以及客观的物理作用力的合力,因此对个体层面来说,是符合行人运动的本质的。社会力模型是连续性的,所以能够精确描述各种层次的作用力,仿真程度是很高的。同时,模型中的行人在行人密度大时能够相互接触产生挤压,符合实际中的拥挤现象,这是元胞自动机等离散性模型所不具备的。

我们在实际运行社会力模型时,发现了一些不足,并初步提出了改进方法:

(1) 模型缺少明确有效的机制来防止行人“重叠”。如果模型采用很大的人体弹性系数来避免行人“重叠”,如 $k = 1.2 \times 10^5 \text{N} \cdot \text{m}^{-1}$,使行人成为非常刚性的结构,以人体压缩量 5cm 为例,接触力会达到 6000N ,这个力将使人瞬时加速到很大的速度,与实际是不符合的,因此模型参数选取需要进一步符合实际。我们在使用`MATLAB`模拟过程中也遇到了这样的问题,在出口处人群拥挤时,重叠现象会使人的速度完全不可控,甚至出现瞬间飞出墙外的情况。当行人距离很近时,模拟的行人表现出与实际不符的振动行为和排斥行为,就像台球一样,一个行人的异常速度会引起连锁反应,而导致一片行人被冲撞开来,这是与实际不符的,在高密度下,人群碰撞会导致摔倒的发生,人的速度会在瞬间减少为0,而不会被撞开并以一个不合常理的高速移动。因此,要将行人运动变得更加连续,即使行人运动平滑。很多模拟中用小的时间步长,来实现行人运动的连续性,但实际情形却又并非如此,因为行人每秒钟不可能走太多步,故可考虑实际的步行、跟随等行为。因此我们设定了行人期望速度的最大值,将行人运动速度限制在最大值之下,行人运动就可以被控制。

(2) 模型中的行人的主观行为只体现在方向始终指向出口的自驱动力和心理排斥力,假设过分简化了行人的路径发现过程,人群疏散信息的传递被简化为从众心理,行人的主观程度也无法体现。因而,模型不能有效模拟行人更复杂的行为。

(3) 算法的复杂性。由于社会力体现了行人之间的相互影响,随着行人数量的增加,运算速度会呈几何级数下降,运行时间过长,因此需要有高效的行人作用范围和数值计算方法,快速找到行人周围与之作用的其他行人。

(4) 模型中障碍物对行人的反作用力采用了与行人之间的作用相似的表达方式,计算方法尚不成熟。实际上行人对障碍物的反应与对其他行人的反应是不同的,对静止障碍物与动态障碍物的反应也不同。因此,社会力模型对复杂环境下的建模很不完善,需分别考虑复杂环境下的各种物体对行人的不同影响,这意味着社会力模型的适用性是有限的,场景越复

杂，社会力模型处理能力就会下降。解决复杂情况下的边界和障碍物可以有两种简化，一是计算最短距离，只考虑和行人距离最近的障碍物或边界对行人的作用力，以距离为函数其他边界或障碍物的作用力视为高阶小量；二是计算最大影响，只考虑对行人作用力最大的边界或障碍物，以力为函数，视其他力为高阶小量。

2. 元胞自动机模型

2.1 元胞自动机简介

上世纪 40 年代，*von Neumann*等人提出了元胞自动机 (*Cellular Automata*, 简称CA) 的概念用于模拟生命系统所特有的自复制现象。元胞自动机最大的特点是时间、空间、状态都离散，每个元胞只取有限多个状态，且其状态改变的规则在时间和空间上都是局部的。

CA是时间与空间都离散的动力学模型，由于其规则的简单，特别适合对复杂系统进行动态模拟。CA模型不同于一般连续性动力系统，其动力学特征不需要使用方程描述，而可以用局部规则来刻画。在该模型中，网格构成的元胞(*cell*)是CA模型的基本元素，一般对于每个元胞，只能取有限个状态值。每个元胞按照所设置的局部规则更新状态，这样通过局部的简单规则就可以涌现出整体的复杂模式。

CA在空间与状态上都是离散的，与连续的行人动力学模型比较有更高的计算效率，而且当对行人运动的规则有准确的认识之后，CA无论在理论研究方面还是在疏散应用上都有比较好的表现。元胞自动机的网格并不限于正四边形，还有正六边形的网格以及不规则形状的网格。四边形的网格简单与直观，特别适用于现有计算机环境下直接进行编程表达；四边形网格对角线无法准确模拟出各向同性的现象，而六边形网格则能较好地克服这一问题，但是其缺点是在计算机编程上难以直接实现。

2.2 元胞自动机模型的建立

本文的模型以二维元胞自动机为基础，将疏散空间划分为均匀的正四边形网格，每个网格对应一个元胞并且只能容纳一个元胞。元胞有三种状态：（1）被一个行人占据；（2）被障碍物或边界占据；（3）空置状态。元胞邻域选择为*von Neumann*型，每个元胞处的行人以一定的转移概率向其前、后、左、右四个元胞移动，或者保持静止，如图所示。接下来我们

用地面场的概念来确定转移概率。

	前	
左	静止	右
	后	

	$P_{i,j-1}$	
$P_{i-1,j}$	$P_{i,j}$	$P_{i+1,j}$
	$P_{i,j+1}$	

2.2.1 静态场

行人疏散中应用最为广泛的元胞自动机模型就是地面场模型。地面场主要包括静态场和动态场。静态场描述场景内任一位置到出口处的最近距离，动态场描述行人的行走轨迹及对他人的从众程度。通过调整静态场和动态场参数，可以控制行人对路线的熟悉程度和从众行为的强度。个体根据规则，在缩短与目的地距离和从众程度的双重机制下运动。

动态场和静态场及其相关影响因素共同决定了元胞转移概率，此概率决定了元胞下一时刻运动方向，可描述行人下一步行走目标的选择。中心元胞行人进入单元格 (i, j) 的概率 P_{ij} 的计算公式为：

$$P_{ij} = N \xi_{ij} \eta_{ij} \exp(k_S S_{ij}) \cdot \exp(k_D D_{ij})$$

$$N = \left\{ \sum_i \sum_j [\xi_{ij} \eta_{ij} \exp(k_S S_{ij}) \cdot \exp(k_D D_{ij})] \right\}^{-1}$$

式中： N 为正规化因子； k_S 和 k_D 分别为静态场参数和动态场参数； S_{ij} 和 D_{ij} 分别为静态场场强和动态场场强； ξ_{ij} 和 η_{ij} 为元胞状态参数，确定规则如下：

$$\xi_{ij} = \begin{cases} 0, & \text{有行人} \\ 1, & \text{无行人} \end{cases} \quad \eta_{ij} = \begin{cases} 0, & \text{有障碍物或边界} \\ 1, & \text{无障碍物或边界} \end{cases}$$

静态场参数是对场所内部结构及场所熟悉程度的描述。静态场强 S_{ij} 反映了建筑物内的结构和障碍物的分布，而参数 k_S 反映了行人对环境的熟悉程度或行人的逃生速度。静态场参数的确立分为两种情况：

(1) 无障碍物时， S_{ij} 的计算公式：

$$S_{ij} = N_S [\max(s_{ij}) - s_{ij}]$$

其中 $s_{ij} = \min\{[(i - i_{0km})^2 + (j - j_{0km})^2]^{1/2}\}$ ，式中： N_S 为无量纲化因子，公式为：

$$N_S = \left\{ \sum_i \sum_j [\max(s_{ij}) - s_{ij}] \right\}^{-1}; \quad s_{ij} \text{为}(i, j) \text{处元胞到第} k \text{个出口的最短距离}; i_{0km} \text{和} j_{0km} \text{分别}$$

为第 k 个出口所包含的第 m 个元胞的坐标。

(2) 有障碍物时, 使用*Dijkstra*算法找出最短路径, 再采用无障碍物下的公式计算静态场强。

2.2.2 动态场

Kircher(2002)受生物信息素的启发引入了动态场, 动态场参数是对从众行为的描述。动态场强 D_{ij} 反映了所有经过 (i,j) 处动态人员吸引力, 参数 k_D 反映行人对其他行人留下的虚拟路径信息的了解或从众心理的程度。

行人通过玻色子来表示该行人留下的虚拟轨迹。玻色子对其他行人有吸引的作用, 而且多个玻色子可以同时占据同一个元胞。当行人从当前位置 (i,j) 移动到其他位置时, 在位置 (i,j) 处留下一个玻色子, 可以表示为 $D_{ij} \rightarrow D_{ij} + 1$, 其中 D_{ij} 为 (i,j) 处的玻色子个数, 即为该处的动态场强。玻色子具有扩散与衰减的机制, 分别以 γ 与 δ 表示扩散与衰减系数($\gamma, \delta \in [0,1]$), 分别表示表示动态场强随时间衰减和向周围元胞扩散的概率, 它反映了行人所留下的路径信息随时间扩散和衰减的过程; 在一个时间步中每个玻色子都以概率 δ 衰减(消失)并以概率 γ 移动到邻域, 具体算法可以为: 对于每个动态玻色子, 抽取之间 $[0,1]$ 均匀分布的随机数 x , 如果 $x < \delta$, 则把该玻色子去除; 否则, 抽取 $[0,1]$ 之间均匀分布的随机数 y , 若 $y < \gamma$, 则把该玻色子移动到随机选择的邻域中。利用平均的思想直接计算:

在 $t \rightarrow t + 1$ 过程中, 分为两步 $t \rightarrow t + 0.5$ 和 $t + 0.5 \rightarrow t + 1$, 考虑为先衰减再扩散。

对于衰减过程: $D_{ij}^{t+0.5} = D_{ij}^t - \delta D_{ij}^t$

对于扩散过程: $D_{ij}^{t+1} = D_{ij}^{t+0.5} - \gamma D_{ij}^{t+0.5} + \frac{\gamma}{4}(D_{i-1,j}^{t+0.5} + D_{i+1,j}^{t+0.5} + D_{i,j-1}^{t+0.5} + D_{i,j+1}^{t+0.5})$

两个过程合并: $D_{ij}^{t+1} = (1 - \delta)(1 - \gamma)D_{ij}^t + \frac{\gamma(1-\delta)}{4}(D_{i-1,j}^t + D_{i+1,j}^t + D_{i,j-1}^t + D_{i,j+1}^t)$

改变衰减与扩散的顺序, 上式不发生变化, 可以看出动态场本质上描述了疏散信息的扩散过程。综上 D_{ij} 的计算步骤如下:

Step1: 初始化 $D_{ij} = 0$;

Step2: 当有 1 人从 (i,j) 元胞处经过, 则此处动态场强增加 ΔD_{ij} , 规定 $\Delta D_{ij} = 1$;

Step3: 根据衰减和扩散规律计算 d_{ij}^t :

$$d_{ij}^t = (1 - \delta)(1 - \gamma)D_{ij}^t + \frac{\gamma(1-\delta)}{4}(D_{i-1,j}^{t-1} + D_{i+1,j}^{t-1} + D_{i,j-1}^{t-1} + D_{i,j+1}^{t-1}) + \Delta D_{ij}^t$$

其中: d_{ij}^t 为未经无量纲化处理的动态场强值, $D_{i-1,j}^{t-1}$, $D_{i+1,j}^{t-1}$, $D_{i,j-1}^{t-1}$, $D_{i,j+1}^{t-1}$ 分别为上一

时刻 4 个邻域动态场强值；

Step4: 对 d_{ij}^t 进行无量纲化处理得 D_{ij}^t ，有 $D_{ij}^t = d_{ij}^t / \sum_i \sum_j d_{ij}^t$ 。

从以上的定义、算法以及平均场近似的关系可以看出玻色子被当作完全相同的粒子，所以很可能出现个体被自己留下的玻色子所吸引（或排斥）的情况。直接使用算法时很容易对玻色子不做区分，而把玻色子认为同质粒子。这样处理会使个体受到自己所释放的玻色子的影响，往往会阻碍个体的运动。为了消除个体在上一步留下的玻色子对自己的影响，C.Burstedde(2001)等提出了修正算法：

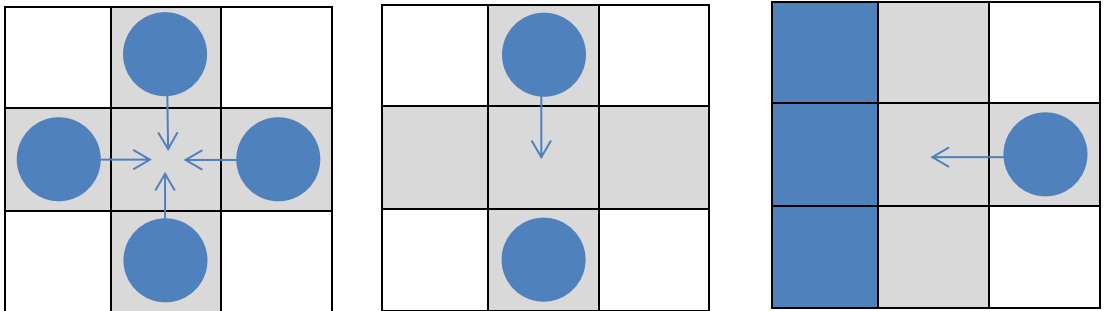
$$d_{ij}^t(k) = d_{ij}^t - I_{(i^t, j^t)_k = (i^{t-1}, j^{t-1})_k}$$

其中： $I_{(i^t, j^t)_k = (i^{t-1}, j^{t-1})_k}$ 为指示函数 $(i^t, j^t)_k$ 和 $(i^{t-1}, j^{t-1})_k$ 分别为行人 k 在 t 和 $t-1$ 时刻的位置，两者相等时指示函数 I 为 1，否则为 0。该方法仅修正一步，衰减较慢时效果不够理想。为了完全避免这些影响，程远(2012)把玻色子按照不同的逃生个体进行区分，用 $D_{ij}^t(k)$ 表示 t 时刻在位置 (i, j) 上个体 k 留下的玻色子个数， $D_{ij}^t(-k)$ 表示除个体 k 之外的其他行人留下的玻色子个数。在区分玻色子后便可以消除影响。

2.2.3 排斥、摩擦和惯性行为

行人疏散过程中人与人、人与建筑之间的相互作用非常复杂，涉及到人的心理和行为特征、人群特征和环境特征等多种因素，有社会力模型的启发，作用力总是可以分3类：摩擦力、排斥力和吸引力，通过对这3类力的基本性质的描述就可以研究行人疏散的基本性质。地面场模型考虑的是“吸引力”，而没有考虑排斥和摩擦，我们利用宋卫国, 于彦飞(2005)的方法引入排斥和摩擦。

(1) 对于静态场和动态场，行人在每次迭代更新之后，首先计算得出周围四个元胞的 P_{ij} ，此时考虑摩擦和排斥的影响修正 P_{ij} 。行人疏散中存在排斥力和摩擦力的几种情况如下：



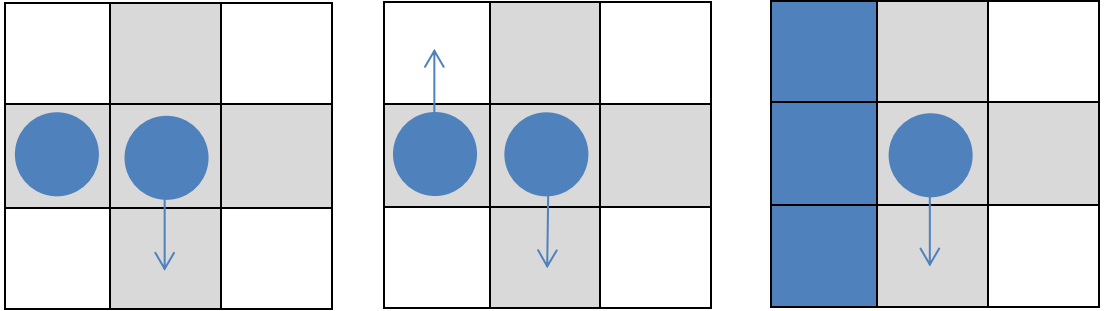
第一种排斥存在情况，箭头代表四个行人的 P_{ij} 最大邻域。此时产生冲突，每个行人的运动概率： $p_i = (1 - r_1)/n$ ($i = 1, 2, \dots, n$)，其中， n 为冲突涉及人数， n 个行人以概率 r_1 全部保持不动，以概率 $1 - r_1$ 运动，每个人被选中进入该元胞的概率相等，一旦某个行人朝着竞争元胞运动，其他未被选中的就在该时间步保持静止。这种情况也被称之为竞争行为。

第二种和第三种运动概率与第一种类似讨论有： $p = 1 - r_2$ ， $p = 1 - r_3$ ，其中 r_2 表示第二种情形中的排斥概率， r_3 表示第三种情形。排斥的出现归因于行人躲避伤害的行为，大小取决于人与人、人与墙的相对速度，相对速度越大，可能受到的伤害就越大，行人就越倾向于躲避。排斥概率随着相对速度的增加单调增加。同时，人体对碰撞有个承受极限，一旦超过这个极限，人总是趋向于躲避，排斥概率应渐近地趋向1。引用人工神经网络中的Sigmoid函数来描述排斥概率：

$$r = k_r \frac{1 - e^{-\alpha v}}{1 + e^{-\alpha v}}$$

一般的Sigmoid函数的解析式为： $S(x) = \frac{1}{1 + e^{-x}}$ ，常被用作神经网络的阈值函数，将 x 映射到(0,1)。

另一部分是摩擦，行人密集时，存在接触现象，元胞自动机无法表述行人接触这一情况，因此需要用摩擦概率来刻画这种实际情况下存在的作用，分别是行人与行人，行人与墙或障碍物的接触减速行为。下面三个图表示存在摩擦的情况：



给出运动概率 p_f ，摩擦概率 f_i ， $p_f = 1 - f_i$, $i = 1, 2, 3$ 。 f_i 取决于人与人、人与墙之间的接触程度、相对速度和摩擦系数，在CA模型中，元胞的尺度是均匀的，接触程度是一样的。因此有：

$$f = k_f \theta v$$

其中， r 和 f 分别为排斥和摩擦概率，即行人在上述情况下分别以概率 r 和 f 静止； k_r 和 k_f 为调节影响大小的参数； α 和 θ 分别为硬度系数和摩擦系数， α 依赖行人的承受极限，

$\alpha \in [0, \infty]$, $\theta \in [0, 1]$; v 为相对速度。

(2) 惯性行为描述：行人移动过程中有惯性因素，运动方向上的转移概率应该更大。在计算目标选择概率时加入惯性作用影响参数 $\omega > 1$ ，即下一时刻目标的运动方向上的邻域内的元胞迁移概率要放大 ω 倍。

2.2.4 行人速度

行人疏散中个体的异质性应该在模型中体现，行人速度是异质性的核心。在地面场模型中可以通过静态场参数 k_s 来控制速度，但是 k_s 本身还反映了行人的逃生动机和对环境的熟悉程度，所以要用其他方法来设置速度。目前设置行人不同速度的方法有两类：一是改变行人的刷新时间（时间步长 Δt ）。若行人有 n 种速度 v_1, v_2, \dots, v_n ，其中最大值为 v_{max} ， Δx 为元胞尺寸，则时间步长 $\Delta t = \Delta x / v_{max}$ 。令 $\lambda_i = v_{max} / v_i$ ，则对速度为 v_i 的行人，每 λ_i 次时间步长更新一次状态。二是改变行人每次刷新所移动的步数，考虑 $v_{max} > 1$ 的情形。第 k 个行人的移动路径可以表示为 $T^k(t) = \{T_0^k(t), T_1^k(t), \dots, T_{v_{max}}^k(t)\}$ ，包括 $v_{max} + 1$ 个元胞。用算子 M 表示从前一位置到下一位置，则有 $T_i^k(t) = M(T_{i-1}^k(t))$ ，其中 $T_i^k(t)$ 是 $T_{i-1}^k(t)$ 的邻域。因此对速度为 v_{max} 的行人，用算子 M 连续作用 v_{max} 次。但是考虑到行人速度的异质性会给代码实现带来很大的难度，因此我们简化了行人速度的设置，忽略了异质性。

2.3 元胞自动机模型参数选取和MATLAB仿真

参考密集人流中典型的人员空间分配标准并考虑到疏散时的拥挤情况，每个元胞定义为 $0.4m \times 0.4m$ 的正方形空间。根据相关文献，我们设置静态场参数 $k_s = 1$ ，动态场参数 $k_d = 0.5$ ， $k_r = k_f = 0.5$ ， $\theta = 0.1$ （行人之间）， $\theta = 0.3$ （行人与障碍物）， $\alpha = 1$ （行人之间）， $\alpha = 1.5$ （行人与障碍物），惯性系数 $\omega = 1.2$ ，扩散和衰减系数 $\gamma = \delta = 0.2$ ，设定期望疏散速度 $v = 2.0m/s$ ，则一个时间步长为 $0.2s$ 。

基于以上建立的模型，编写MATLAB程序，房间设置为 $8m \times 12m$ ，出口设置为4个排列在边界的元胞，设置50个行人，模拟情况如下。Figure 5为初始状态行人分布，Figure 6为行人形成的聚集。

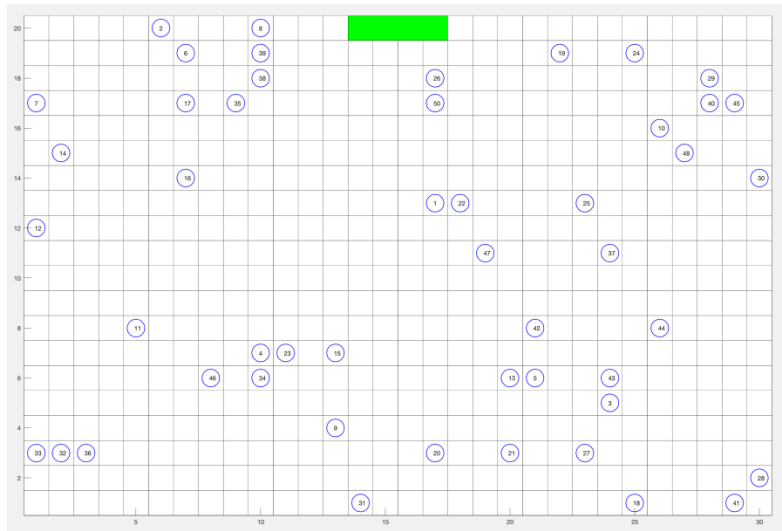


Figure 5 初始元胞分布

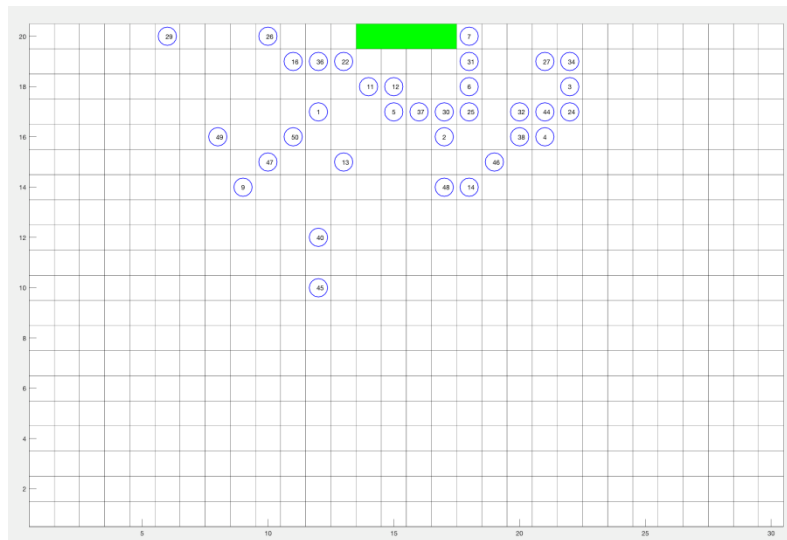
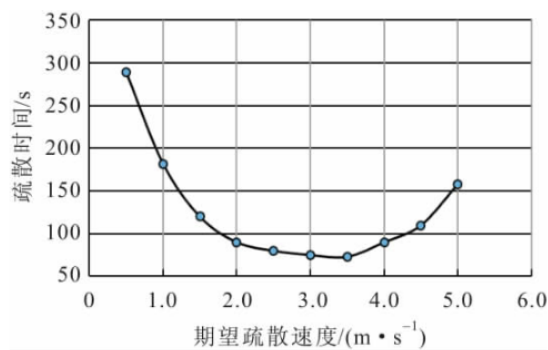


Figure 6 元胞聚集状态

由于我们设置了一致的行人速度,忽略了异质性,为了得到行人在不同期望速度下全部疏散时间的关系,我们需要设置不同的时间步长 Δt ,进一步因为行人速度 v 的改变,我们模型中的 k_r, k_f 需要不断随着调整,我们尚没有得到满意的结果,但郭良杰,赵云胜(2014)设置了同样的房间,初始人数设为100人的情况下,得到了比较好的结果,如下图:

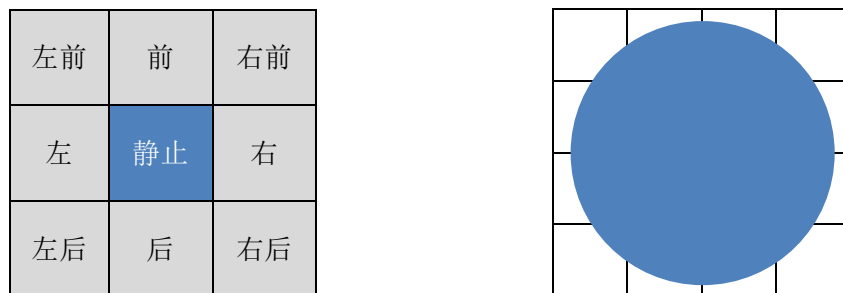


如果我们同样考虑不同出口大小对全部疏散时间的影响，由于出口大小只能是 Δx 的整数倍，因此必须以 $0.4m$ 为间隔用来获得数据的出口宽度值过少，无法得到有一定精度的出口宽度与全部疏散时间的关系。进一步我们通过调整静态场参数 k_S 和动态场参数 k_D ，来研究改变行人对环境的熟悉程度和行人的从众程度对平均疏散时间的影响，分析后可以得出平均疏散时间 $\bar{T} \propto k_S$ ， \bar{T} 随着 k_D 的增大先减小后增大，这是因为过度的从众会导致行人形成拥挤的队列流，造成疏散路径和疏散出口都没有被合理利用，降低了疏散效率。在 k_S 不大时，适度的从众行为会缩短疏散时间。

2.4 元胞自动机模型的优势、不足和改进

CA模型的优势是模型的简单性以及由此得到的高效的计算能力。我们从模型假设来一步一步分析CA模型的不足之处。

首先，行人在模型中被描述为占据一个元胞，尺寸为 $0.4m \times 0.4m$ ，以这种尺寸计算行人密度，最高为 $6.25 person/m^2$ ，但是在密度更大时，行人仍有运动，此时CA模型就受到了限制。同时，在规定行人移动方向时，我们假定了行人只能往前、后、左、右移动，运动方向并不完全符合实际。进一步的，我们可以考虑Moore型邻域，行人运动方向可以为 45° 的整数倍，进一步优化模型的精度，如图：



我们可以考虑多元胞的CA模型，即一个行人占据一个 4×4 的网格，这样我们可以考虑行人之间因为挤压所产生的力，用重叠的元胞个数来刻画力的大小，元胞位置来刻画力的方向，但是对于时间和空间都离散的CA模型，加入连续性的力是不够融洽的，这时的元胞更新规则更为复杂。

与社会力模型相比，CA模型付出一些精度缺失的代价。模型的精度有所下降，同时行人间的个体差异性无法在模型中表达出来。我们使用概率来描述行人下一时刻的移动方向，很好的描述了行人运动的不确定性，但是在人数很少的情况下，行人会出现左右摇摆的情况，不能够及时走出房间，这与实际也是不符合的。使用概率的描述在多次迭代之后，行人确实

能走出房间，但其运动时间就与正常疏散时间有了较大偏差。

行人的元胞尺寸与模拟的空间精度相关，可以描述的物体尺寸为元胞尺寸 Δx 的整数倍。若 Δx 取值过大，就得不到空间尺度上的精确分析。行人速度 $v = \Delta x / \Delta t$ ，则 $\Delta t = \Delta x / v$ ，若需要更小时间尺度的模拟，就需要更小空间步长 Δx 。这时行人就不得不占据多个格子，因此CA模型正是因为行人与元胞的契合而变得优异，也正是因为要与行人尺寸匹配而变得乏力。在复杂情形下，静态场强的计算比受力分析来得容易得多。

在存在楼梯等连接层结构的情况下，CA模型可以增加一个高度维度，来刻画静态场参数，但是处理动态场会变得非常复杂。在体育场等斜面形空间，CA模型仍能使用，与楼梯结构相比，斜面结构更为光滑，考虑高度便可以进行场强的计算，因为实质上，斜面结构仍然是二维平面。CA模型的改进会随着更新规则的复杂变得越来越精细，从而失去了原本CA模型由简单更新规则得到复杂现象的突出特点。

综合上述分析，我们可以明白，社会力模型作为连续的精细的模型，运算是很复杂的，近一步的优化方向是优化数值计算方法，模型本身可以更完善的地方已经不多；元胞自动机模型作为离散的粗糙的模型，运算效率很高，近一步的优化则使模型的假设和构成更为复杂。行人疏散问题是一个复杂的问题，用数学方法来模拟这种实际问题，会遇到很多困难，适度的简化会帮助我们更好的描述行人运动的本质。虽然我们的讨论和分析还有很多不完善的地方需要进一步严格化，不过我们认为以上内容作为对行人疏散问题的初步建模是已经足够的。

参考文献

- [1]Weidmann U.Transporttechnik der Fugnger[J].Schriftenreihe IVT-Berichte,1993.
- [2]Helbing D, Molnar P.Social Force Model for Pedestrian Dynamics[J],1995, Physical Review E,51,4282-4286.
- [3]Helbing D,Farkas I,Vicsek T.Simulating Dynamical Features of Escape Panic[J]. Nature,2000,407(6803):487-490.

- [4]Daamen W, Hoogendoorn S P. Modeling pedestrian flows through transfer stations [C]. Transportation Research Board Annual Meeting, 2002.
- [5]Helbing D, Farkas I J, Molnar P, Vicsek T. Simulation of Pedestrian Crowds in Normal and Evacuation Situations Normal and Evacuation Situations[J]. Pedestrian and Evacuation Dynamics, 2002(2):21-58.
- [6]Lakoba T I, Kaup D J, Finkelstein N M. Modifications of the Helbing-Molnar-Farkas-Vicsek Social Force Model for Pedestrian Evolution[J]. Simulation, 2005, 81(5):339-352.
- [7]单庆超,张秀媛,张朝峰.社会力模型在行人运动建模中的应用综述[J]. 城市交通, 2011, 9(06):71-77.
- [8]嵇保玲. 基于社会力的同伴群走行模型研究[D]. 北京交通大学, 2014.
- [9]von Neumann, J. Theory of Self-reproduction Automata[M]. Urbana, IL:University of Illinois Press, 1966.
- [10]Kircher A, Schadschneider A. Simulation of evacuation process using a bionics-inspired cellular automaton model for pedestrian dynamics[J]. Physica A: Statistical Mechanics and Its Applications, 2002, 312(1-2):260-276.
- [11]Burstedde C, Klauck K, Schadschneider A, et al. Simulation of pedestrian dynamics using a two-dimensional cellular automaton[J]. Physica A: Statistical Mechanics and Its Applications, 2001, 295(3-4):507-525.
- [12]宋卫国, 于彦飞, 范维澄, 等. 一种考虑摩擦与排斥的人员疏散元胞自动机模型[J]. 中国科学-E 辑-工程科学-材料科学, 2005, 35(7):725-736.
- [13]于彦飞. 人员疏散的多作用力元胞自动机模型研究[D]. 合肥:中国科学技术大学, 2008.
- [14]程远. 基于演化博弈论的群体疏散行为研究[D]. 北京:北京化工大学, 2012.
- [15]郭良杰, 赵云胜. 基于元胞自动机模型的人员疏散行为模拟[J]. 安全与环境工程, 2014, 21(04):101-106.

附录A

社会力模型部分 $MATLAB$ 代码

```
%% 初始化
clear

global X
global Y
global D
global desti
global tau

global A
global B
global k
global K
global A_
global B_
global k_
global K_

global V_EXP

N_init = 50;

X = 12; Y = 8;

V_EXP = 2;

D = 1.6;

NO = (1:N_init)' ; % 行人序号

SIZE = normrnd(50, 3.3, N_init, 1)/2/100; % 行人尺寸

M = normrnd(65, 5, N_init, 1); % 行人质量

COORD = rand_COORD(SIZE); % 行人坐标

F = zeros(N_init, 2); % 行人受力

V = zeros(N_init, 2); % 行人速度
```

```

N = rand(N_init, 1)/2; % 行人从众系数

desti = [X/2, Y]; % 目标点

tau = 0.5;

% 人-人作用力参数
A = 2.1; B = 0.08;
k = 4e4; K = 6e4;

% 人-墙作用力参数
A_ = 10; B_ = 0.3;
k_ = 4e4; K_ = 6e4;

%% 运算
T = 1000;
delta = 0.0001; % 时间步长
t = 0;

fig = figure(1);
while t < T
    % 判断是否有人逃生（进入以门为直径的半圆内）
    p = 1;
    while p <= size(COORD,1)
        if normest( COORD(p, :)-desti ) < D/2

            COORD(p, :) = []; SIZE(p, :) = [];
            V(p, :) = []; F(p, :) = [];

            M(p) = []; N(p) = []; NO(p) = [];

            continue
        end

        p = p + 1;

    end

    if size(COORD,1) == 0
        break
    end
end

```



```

F_self = p_self( COORD, V, M, N );
F_p2p = p_2_p(COORD, SIZE, V, M);
F_p2B = p_2_B(COORD, SIZE, V, M);

F = F_self + F_p2p + F_p2B;

V_end = V + F./ M * delta;

for p = 1:size(COORD,1)
    if normest(V_end(p,:)) > 2*V_EXP
        V_end(p,:) = V_end(p,:)/normest(V_end(p,:))* 2*V_EXP;
    end
end

COORD = COORD + (V + V_end) / 2 * delta;

V = V_end;

t = t + delta;

if mod(round(10000*t),1000) == 0
    delete(fig)
    fig = plot_social(COORD, SIZE, V, NO, F_self);
    pause(0.1)
end

end

t

' the end'

% 人-人作用力
function F_p2p = p_2_p(COORD, SIZE, V, M)
    global A
    global B
    global k
    global K

    N_p = size(COORD,1);
    F_soc = zeros(N_p, 2);
    F_ph = zeros(N_p, 2);

```

```

for p = 1: N_p
    f_soc = zeros(1, 2);
    f_ph = zeros(1, 2);

    for q = 1: N_p
        if q == p
            continue
        end

        r_ab = SIZE(p)+ SIZE(q);
        d_ab = normest(COORD(p, :)-COORD(q, :));

        if d_ab <= B+ r_ab
            n_ab = (COORD(p, :)-COORD(q, :))/d_ab;
            t_ab = [n_ab(2), -n_ab(1)];
            delta_v = (V(q)- V(p))*t_ab;

            f_soc = f_soc+ M(p)* A* exp( (r_ab - d_ab)/B )* n_ab ;
            f_ph = f_ph+ k* relu(r_ab- d_ab)* n_ab+ ...
                K* relu(r_ab - d_ab)* (delta_v* t_ab')* t_ab;
        end

    end

    F_soc(p, :) = f_soc;
    F_ph(p, :) = f_ph;
end

F_p2p = F_soc+ F_ph;

end

% 行人自驱力
function F_self = p_self( COORD, V, M, N )
    global desti
    global tau
    global X
    global Y
    global V_EXP

    N_p = size(COORD, 1);
    F_self = zeros(N_p, 2);

```

```

rho = N_p/(X*Y);
v_exp = V_EXP*(1-exp(-1.913*(1/rho-1/5.4)));

for p = 1:N_p
    e = (desti - COORD(p, :))/normest(desti - COORD(p, :));

    if normest(sum(V)) == 0
        e_others = zeros(1,2);
    else
        e_others = sum(V)/normest(sum(V));
    end

    e_correct = (1-N(p))* e + N(p)* e_others;
    e_correct = e_correct/ normest(e_correct);

    F_self(p, :) = M(p)* (v_exp* e_correct - V(p, :))/ tau ;

    if sum(sum(isnan(F_self)))>0
        pause
        error('F_self_error');
    end
end

end

```

附录B

元胞自动机部分*MATLAB*代码

```

%% 主程序
% initialize
clear all
global k_s
global k_D

global k_r
global alpha
global alpha_

global k_f
global theta
global theta_

```

```

global V_exp

global S_field
global r
global beta

global X
global Y
global d
global w

k_r = 0.5; k_f = 0.5;

theta = 0.1; theta_ = 0.3;

alpha = 1; alpha_ = 1.5;

p = 0.05;
beta = 0.2;
r = 0.2;

V_exp = 2; % 期望速度

N_init = 50; % 初始人数

l = 0.4; % 元胞边长
X = 12/l; Y = 8/l;

d = 1.6/l; % 出口格点数
D = [ [(X/2-d/2+1):(X/2+d/2)]', Y*ones(d, 1) ]; % 出口位置

%% 计算
% 随机生成行人坐标
NO = (1:N_init)';
p = randperm(X*Y, N_init)';
COORD = [p- X* (ceil(p/X)-1), ceil(p/X)];
% clear p N_init

w = 1.2; % 惯性系数
W = 5 * ones(N_init, 1); % 代表行人方向
% 1-(i, j+1); 2-(i, j-1); 3-(i+1, j); 4-(i-1, j); 5-(i, j)

% 生成静态场场强

```

```

k_s = 1;
S_field = zeros(X, Y);
for i = 1:X
    for j = 1:Y
        S_field(i, j) = normest( min(abs(D-[i, j])) );
    end
end
M = max(max(S_field));
S_field = M-S_field;

% 刻画动态场
k_D = 0.1;
D_field = zeros(X, Y);

T = 1000;
delta = 0.2;
t = 0;

while t<T
    % 判断是否有人离开
    l = 1;
    outer = [];
    while l <= size(COORD,1)
        if sum( ismember(D, COORD(l,:), 'rows') ) == 1
            outer = [outer; COORD(l,:)];

            COORD(l,:) = [];
            W(l,:) = [];
            NO(l,:) = [];

            continue
        end

        l = l+1;
    end

    [fig] = plot_cell(COORD, NO);

    N_p = size(COORD,1); % 现有人数

    if N_p == 0
        break
    end
end

```

```

[next_COORD, W] = next_coord(COORD, W, D_field);

D_field = D_field_upgrade(D_field, COORD, W, outer);

COORD = next_COORD;

pause(1)
delete(fig)

t = t + delta;

end

% 动态场更新
function next_D_field = D_field_upgrade(D_field, COORD, W, outer)
    global X
    global Y
    global r
    global beta

    next_D_field = (1-r) * (1-beta) .* D_field;
    N_p = size(COORD,1);

    for p = 1:N_p
        if W(p) ~= 5
            next_D_field(COORD(p,1), COORD(p,2)) = next_D_field(COORD(p,1),
COORD(p,2)) + 1;
        end
    end

    for p = 1:size(outer, 1)
        next_D_field(outer(p,1), outer(p,2)) = next_D_field(outer(p,1),
outer(p,2)) + 1;
    end

    next_D_field(1,1) = next_D_field(1,1) + beta* (1-r)/2 * (D_field(1,2)+
D_field(2,1));
    next_D_field(X,1) = next_D_field(X,1) + beta* (1-r)/2 * (D_field(X-1,1)+
D_field(X,2));
    next_D_field(1,Y) = next_D_field(1,Y) + beta* (1-r)/2 * (D_field(2,Y)+
D_field(1,Y-1));
    next_D_field(X,Y) = next_D_field(X,Y) + beta* (1-r)/2 * (D_field(X-1,Y)+

```

```

D_field(X,Y-1));

    for i = 2:X-1
        next_D_field(i,1) = next_D_field(i,1) + beta*(1-r)/3 *
(D_field(i-1,1)+D_field(i+1,1)+D_field(i,2));
        next_D_field(i,Y) = next_D_field(i,Y) + beta*(1-r)/3 *
(D_field(i-1,Y)+D_field(i+1,Y)+D_field(i,Y-1));
    end

    for j = 2:Y-1
        next_D_field(1,j) = next_D_field(1,j) + beta*(1-r)/3 *
(D_field(1,j-1)+D_field(1,j+1)+D_field(2,j));
        next_D_field(X,j) = next_D_field(X,j) + beta*(1-r)/3 *
(D_field(X,j-1)+D_field(X,j+1)+D_field(X-1,j));
    end

    for i = 2:X-1
        for j = 2:Y-1
            next_D_field(i,j) = next_D_field(i,j) + beta*(1-r)/4 *
(D_field(i,j-1)+D_field(i,j+1)+D_field(i-1,j)+D_field(i+1,j));
        end
    end

    next_D_field = next_D_field ./ sum(sum(next_D_field));

end
% 生成下一步坐标
function [next_COORD, W] = next_coord(COORD, W, D_field)
    global k_s
    global k_D
    global S_field
    global w

    N_p = size(COORD,1);
    pre_next_COORD = zeros(N_p, 2);

    for p = 1:N_p
        now_coord = COORD(p, :);
        % 1-(i,j+1);2-(i,j-1);3-(i+1,j);4-(i-1,j);5-(i,j)
        P = ones(5,1);

        near_coord = now_coord + [[0,1];[0,-1];[1,0];[-1,0];[0,0]];

```

```

P(5) = exp( k_s* S_field(now_coord(1), now_coord(2)) ) *...
      exp( k_D* D_field(now_coord(1), now_coord(2)) );

IS = is_near_wall(now_coord) + is_near_p(now_coord, COORD);

for k = 1:4

    if IS(k) == 1
        P(k) = 0;
        continue
    end

    P(k) = exp(k_s*S_field(near_coord(k,1), near_coord(k,2))) *...
          exp(k_D*D_field(near_coord(k,1), near_coord(k,2)));

    if W(p) == k
        P(k) = w * P(k);
    end

end

P = P/sum(P);

epsilon = rand(1);
for k = 1:5
    epsilon = epsilon - P(k);
    if epsilon <= 0
        W(p) = k;
        break
    end
end

pre_next_COORD(p, :) = near_coord(k, :);
end

[pre_next_COORD, W] = repel(COORD, pre_next_COORD, W);

[pre_next_COORD, W] = friction(COORD, pre_next_COORD, W);

[next_COORD, W] = compete(COORD, pre_next_COORD, W);

end

```