

Tese de Doutorado

Modelos de Evacuação Emergencial via Autômatos Celulares com Mudança de Rotas e Campo de Grupos

por

Leandro Alves Pereira

Orientador:

Luiz Henrique Duczmal

Coorientador:

Frederico R. B. Cruz

Julho de 2016

LEANDRO A. PEREIRA

**MODELOS DE EVACUAÇÃO EMERGENCIAL VIA
AUTÔMATOS CELULARES COM MUDANÇA DE
ROTAS E CAMPO DE GRUPOS**

Belo Horizonte
2016

LEANDRO A. PEREIRA

**MODELOS DE EVACUAÇÃO EMERGENCIAL VIA
AUTÔMATOS CELULARES COM MUDANÇA DE
ROTAS E CAMPO DE GRUPOS**

Tese apresentada ao Departamento de Estatística do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais, como requisito parcial para a obtenção de Título de Doutor em Estatística.

Luiz Henrique Duczmal Orientador:

Luiz Henrique Duczmal

Co-orientador:

Frederico R. B. Cruz

Belo Horizonte
2016



UNIVERSIDADE FEDERAL DE MINAS GERAIS

PROGRAMA DE PÓS-GRADUAÇÃO EM ESTATÍSTICA

UFMG

ATA DA DEFESA DE TESE DO ALUNO LEANDRO ALVES PEREIRA

Realizou-se, no dia 05 de julho de 2016, às 14:00 horas, 2076 - ICEx, da Universidade Federal de Minas Gerais, a 41ª defesa de tese, intitulada "*Modelos de Evacuação Emergencial via Autômatos Celulares com Mudança de Rotas e Campo de Grupos*", apresentada por LEANDRO ALVES PEREIRA, número de registro 2013655368, graduado no curso de ESTATÍSTICA, como requisito parcial para a obtenção do grau de Doutor em ESTATÍSTICA, à seguinte Comissão Examinadora: Prof. Luiz Henrique Duczmal - Orientador (DEST/UFMG), Prof. Frederico Rodrigues Borges da Cruz - Coorientador (DEST/UFMG), Profª. Denise Burgarelli Duczmal (MAT/UFMG), Prof. Roberto da Costa Quinino (DEST/UFMG), Prof. Fernando Luiz Pereira de Oliveira (DEST/UFOP), Profª. Rosana Sueli da Motta Jafelice (FAMAT/UFU).

A Comissão considerou a tese:

Aprovada

Reprovada


Finalizados os trabalhos, lavrei a presente ata que, lida e aprovada, vai assinada por mim e pelos membros da Comissão.
Belo Horizonte, 05 de julho de 2016.


Prof. Luiz Henrique Duczmal (Doutor)


Prof. Frederico Rodrigues Borges da Cruz (Doutor)


Profª. Denise Burgarelli Duczmal (Doutora)


Prof. Roberto da Costa Quinino (Doutor)


Prof. Fernando Luiz Pereira de Oliveira (Doutor)


Profª. Rosana Sueli da Motta Jafelice (Doutora)



UNIVERSIDADE FEDERAL DE MINAS GERAIS

PROGRAMA DE PÓS-GRADUAÇÃO EM ESTATÍSTICA

UFMG

FOLHA DE APROVAÇÃO

"Modelos de Evacuação Emergencial via Autômatos Celulares com Mudança de Rotas e Campo de Grupos"

LEANDRO ALVES PEREIRA

Tese submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em ESTATÍSTICA, como requisito para obtenção do grau de Doutor em ESTATÍSTICA, área de concentração ESTATÍSTICA E PROBABILIDADE.

Aprovada em 05 de julho de 2016, pela banca constituída pelos membros:

Prof. Luiz Henrique Duczmal - Orientador
DEST/UFMG

Prof. Frederico Rodrigues Borges da Cruz - Coorientador
DEST/UFMG

Prof. Denise Burgarelli Duczmal
MAT/UFMG

Prof. Roberto da Costa Quinho
DEST/UFMG

Prof. Fernando Luiz Pereira de Oliveira
DEST/UFOP

Prof. Rosana Sueli da Motta Jafelice
FAMAT/UFU

Belo Horizonte, 5 de julho de 2016.

Dedico este trabalho a todos os estudantes que sonham com um país melhor, que para isso não medem esforços nos seus estudos e que acreditam que a educação é a melhor forma de transformar um país.

AGRADECIMENTOS

Finalmente o último passo foi dado. Passo este não menos importante que todos os outros anteriores, mas que tem um sabor especial porque representa o fechamento de um ciclo que se iniciou há 16 anos atrás. Para alguns jovens brasileiros, a conclusão de um trabalho como este pode ser simplesmente um fato comum, já previsto desde antes do seu nascimento. Para mim é bem diferente, pois venho de uma parcela da sociedade na qual concluir o ensino médio já é considerado uma grande vitória. Cursar uma pós-graduação, principalmente a nível de doutorado, é algo tão inimaginável quanto visitar a lua.

Primeiramente, o meu agradecimento especial não podia deixar de ser para a pessoa que me fez chegar até aqui, até quando eu mesmo pensava em desistir: minha mãe, senhora Luzia Campinho Pereira. Posso dizer que esta vitória é mais dela do que minha. Nunca poupou esforços. Mesmo com condições financeiras restritas, conduziu-me até o ponto em que eu pudesse caminhar sozinho. Sem ela nada disso seria possível. Agradeço a minha esposa, Sueli Paula de Souza Alves, que esteve comigo durante essa caminhada. Ela foi a minha parceira durante grande parte deste processo, sendo extremamente importante em momentos muito delicados, como na mudança de cidade e durante todo o período de estudos pesados. Agradeço aos meus colegas de trabalho da UFU, por suprirem a minha ausência durante o período em que estive cursando o doutorado. Agradeço à minha instituição de trabalho, a UFU, por permitir e conceder afastamento integral para a minha capacitação. Agradeço aos meus familiares pelos momentos que tivemos durante o período em que estive em BH. Com vocês tive momentos de descontração que foram importantíssimos para a minha saúde mental. Agradeço aos meus orientadores, Luiz e Frederico, pelas ideias, correções, incentivos e aprendizagens não só na área profissional, mas para a vida. Sem dúvida foram grandes parceiros nesta jornada, desde a época de mestrado trabalhando juntos.

A todos, os meus mais sinceros agradecimentos. Vocês também fazem parte deste momento tão especial. Esta vitória não é só minha, mas de todos vocês.

RESUMO

Neste trabalho propomos uma extensão dos modelos de autômatos celulares aplicados à evacuação emergencial em dinâmica de pedestres. As novas extensões propostas são a probabilidade de mudança de rotas e o campo de grupos. A primeira extensão permite que, durante as simulações, os pedestres possam mudar de direção em situações particulares, como congestionamentos, acessando uma rota alternativa de saída. A segunda extensão adiciona um campo, cujo objetivo é fazer os pedestres que estão em grupos caminharem sempre próximos, buscando a saída de forma conjunta e tomando decisões comuns para todos os elementos. Vários experimentos foram realizados com o objetivo de estudar o efeito dessas novas extensões nas medidas de segurança, mais precisamente no tempo de evacuação e na contagem de retenções, além de realizar comparações frente aos modelos anteriores. As principais conclusões são que os efeitos dessas novas extensões alteram de forma efetiva as medidas de segurança e portanto podem ser importantes para um aperfeiçoamento dos modelos, que por sua vez, fornecerão melhores estimativas.

Palavras-chave: Evacuação emergencial, Autômatos celulares, Mudanças de rota.

ABSTRACT

In this paper we propose an extension of cellular automata models applied to the emergency evacuation pedestrians dynamics. The new extensions are the route change probabilities and the group fields. The first extension allows pedestrians to change directions when necessary to access an alternative exit route. The second extension adds group fields such that pedestrians always walk close to each other and evacuate together from an environment. Several computational experiments were conducted to study the effect of these new extensions in the security measures, more precisely in the evacuation time and in the number of pedestrians held, and to perform comparisons with another previous models. The main conclusions are that the effects of these new extensions modify effectively the security measures and can therefore be important to improve the models leading to better estimates.

keywords: Emergency evacuation, Cellular automata, Route changes.

LISTA DE ILUSTRAÇÕES

1	Vizinhanças de <i>von Neumann e Moore</i>	16
2	Progressão geométrica de um AC unidimensional de regra 30.	18
3	Concha da espécie <i>Conus Textile</i> com padrão semelhante à regra 30. . .	18
4	Estados no tempo t e tempo $t + 1$ do modelo de Schadschneider (2002). . .	19
5	Representação de deslocamento para a vizinhança.	20
6	Ambiente real e sua representação por um AC.	21
7	Possíveis deslocamentos e a matriz de preferências $M = (M_{ij})$	23
8	Exemplo de CE a partir da contagem de passos	26
9	Exemplo de CE a partir da distância Euclidiana	26
10	Exemplo de posições reservadas para $D_A = 4$	31
11	Calculo de $A_{ij}^{(m')}$ e a relação entre $A_{ij}^{(m')}$ e as direções	31
12	Representação da difusão e decaimento do CD.	34
13	Representação da primeira regra de mudança de rota	45
14	Representação dos CE no cálculo de μ	47
15	Representação da segunda regra de mudança de rota	48
16	Representação do campo de grupo (CG)	50

17	Representação de obstáculo que impossibilita movimentação do grupo .	51
18	Representação de colapso provocado pelo CG	51
19	Ambiente utilizado na segunda etapa de simulações	56
20	Ambiente utilizado na primeira e terceira etapa de simulações	57
21	Foto do ambiente com lotação de 30% em início de simulação.	59
22	Foto do ambiente com lotação de 30% em estágio avançado de simulação. .	59
23	Planilha de criação do autômato.	60
24	Gráficos para o tempo de evacuação	71
25	Gráficos para a contagem de retenções	72

LISTA DE TABELAS

1	Exemplo de AC determinístico.	19
2	Exemplo de AC estocástico.	19
3	Parâmetros para segunda etapa de simulações.	58
4	Parâmetros para terceira etapa de simulações.	58
5	Valores da matriz de preferências.	62
6	Estatísticas descritivas da segunda etapa de simulações.	64
7	Estatísticas descritivas para o tempo de evacuação (T).	69
8	Estatísticas descritivas para a contagem de retenções (CR).	70

LISTA DE ABREVIATURAS E SIGLAS

- AC Autômatos celulares;
- CD Campo dinâmico;
- CE Campo estático;
- CA Campo de antecipação;
- CG Campo de grupo;
- T Tempo de evacuação total;
- CR Contagem de retenções.

LISTA DE SÍMBOLOS

- \mathcal{A} Autômato celular d -dimensional;
- \mathbb{Z}^d Espaço de reticulados de \mathcal{A} com d dimensões;
- S Conjunto finito que representa os estados possíveis das células de \mathcal{A} ;
- N Subconjunto finito de \mathbb{Z}^d , chamado de vizinhança de \mathcal{A} ;
- δ Regra local de transição de \mathcal{A} ;
- $c_t^{\mathcal{A}}$ Configuração ou estado global de \mathcal{A} no instante t ;
- z Ponto local de \mathbb{Z}^d ;
- $N_V(c)$ Vizinhança de *von Neumann*;
- $N_M(c)$ Vizinhança de *Moore*;
- M Matriz final de preferências para vizinhança 1 passo;
- $M^{(a)}$ Matrizes iniciais de preferências para vizinhança 1 passo;
- U Matriz final de preferências para vizinhança 2 passos;
- $U^{(a)}$ Matrizes iniciais de preferências para vizinhança 2 passos;
- M_{ij} Elemento da posição (i, j) da matriz de preferências M ;
- U_{ij} Elemento da posição (i, j) da matriz de preferências U ;
- p_{ij} Probabilidade de transição;

N	Constante normalizadora;
D_{ij}	Força dinâmica da célula (i, j) ;
K_d	Constante multiplicativa da força dinâmica;
S_{ij}	Força estática da célula (i, j) ;
K_s	Constante multiplicativa da força estática;
n_{ij}	Estado da célula (i, j) ;
ξ_{ij}	Identificador de obstáculos da posição (i, j) ;
m	Código de direção preferencial do pedestre;
m'	Código de direções;
$E_{ij}^{(m)}$	Força de antecipação CA para a célula (i, j) ;
$E_{ij}^{(m)}$	Efeito de antecipação do CA;
K_a	Constante multiplicativa da efeito de antecipação;
D_A	Tamanho do campo de antecipação;
D_{ij}^t	Força dinâmica da célula (i, j) no tempo t ;
β	Constante de difusão;
δ	Constante de decaimento;
Q_A	Variável binária para determinação da matriz de preferências;
m_1	Elemento de maior probabilidade na matriz de preferências 1 passo;
u_1	Elemento de maior probabilidade na matriz de preferências 2 passos;

K_r	Constante da força das probabilidades de mudança de rota;
α	Constante de difusão;
W	Numero de CE presentes no autômato;
μ_g	Probabilidade da célula continuar com o mesmo CE;
μ_p	Probabilidade da célula mudar de CE;
q_p	Variável auxiliar no calculo de μ_p e μ_g ;
π	Probabilidade de mudança de rota em congestionamento;
ζ	Ocupação mínima da vizinhança n_z para mudar de rota com probabilidade π ;
δ	Probabilidade de um elemento retido seja deixado pelo grupo;
I_{gl}	Variável <i>dummy</i> que garante que o grupo movimente-se dentro do CG;
w_p	Nome atribuído a um CE, sendo $p = 1, \dots, n$.

SUMÁRIO

1	Introdução	1
2	Fundamentos Teóricos	8
2.1	Modelos de Forças Sociais	9
2.2	Modelos Baseados em Teoria de Filas	10
2.3	Modelos de Sistemas Multiagentes	13
2.4	Autômatos Celulares	14
2.4.1	Conceitos Iniciais	14
2.4.2	Configuração ou Estado Global de \mathcal{A}	15
2.4.3	Vizinhanças	16
2.4.4	Autômatos Celulares Determinísticos e Estocásticos	17
2.5	Autômatos Celulares Estocásticos para Simulação de Pedestres	19
2.5.1	Escala de Células e velocidade	21
2.5.2	Matriz de Preferências	22
2.5.3	Campo Dinâmico (<i>Dynamic Floor Field</i>)	24
2.5.4	Campo Estático (<i>Static Floor Field</i>)	25

2.5.5	Campo de Antecipação (<i>Anticipation Floor Field</i>)	26
2.5.6	Probabilidades de Transição	27
2.5.7	Difusão e Decaimento	33
2.6	Observações finais	35
3	Autômatos Celulares com Mudança de rota	36
3.1	Probabilidades de Transição	38
3.2	Regras para Mudança de Direção	44
3.3	Campo de Grupos	48
3.4	Condições Iniciais	51
3.5	Observações finais	53
4	Simulações e Resultados	54
4.1	Criação do Ambiente	59
4.2	Primeira Etapa de Simulações: Avaliação Subjetiva	60
4.3	Segunda Etapa de Simulações	64
4.4	Terceira Etapa de Simulações	66
4.5	Observações finais	68
5	Conclusões e Trabalhos Futuros	73

Referências 77

Apêndice A - Programa de Simulação 80

1 INTRODUÇÃO

Desde a sua origem, o homem trabalha para entender o mundo em que vive. A partir dessa necessidade, várias teorias sobre os mais diversos fenômenos reais são desenvolvidas e melhoradas à medida que o nosso conhecimento acerca desses fenômenos aumentam. Para organizar e estudar uma parte desses fenômenos surgiu o conceito de *sistemas complexos*, área em que este trabalho se situa. Dizemos que um sistema é um conjunto de elementos interconectados, de modo a formar um todo organizado. Um sistema é dito ser um sistema complexo quando suas propriedades não são uma consequência natural de seus elementos constituintes vistos isoladamente (ILACHINSKI, 2001). Neste caso o sistema é composto por várias partes que se interagem com a habilidade de formar novas qualidades no comportamento coletivo. Também podemos dizer que, em um sistema complexo, o todo é mais que a soma das partes. Como exemplo, podemos destacar o código genético, tráfego de veículos e pessoas, colônia de insetos, evolução de espécies, redes sociais e outras. Durante muito tempo, muitas teorias desenvolvidas para estudos de alguns sistemas complexos exigiam cálculos e processamentos avançados, inviabilizando seu uso. Os potentes computadores acessíveis à maioria dos cientistas nos dias de hoje permitem que as teorias antigas e atuais sejam experimentadas através de modelos, facilitando o nosso entendimento do mundo que

nos rodeia. Segundo Alonso (2008), devido à ampla diversidade de áreas em que estudos desta natureza podem ser aplicados, as abordagens multidisciplinares são as mais frequentes. Os sistemas complexos são caracterizados por propriedades que decorrem em grande parte da relação não linear entre as partes, dentre as quais se destacam a unidade coletiva, organicidade funcional e padrão coletivo ou propriedade emergente (CAMAZINE et al., 2003). Dentre as áreas que trabalham com características complexas e que podem ser estudadas como sistemas complexos temos: redes, teoria de filas, sistemas dinâmicos, percolação, inteligência artificial, autômatos celulares, genética, teoria da evolução e outras. A teoria de autômatos celulares (AC) surge por volta da década de 1940, tendo como seu pioneiro von Neumann (VON NEUMANN, 1966), que foi o primeiro a trabalhar com conjunto de estado e vizinhanças, no qual ele chamou de célula, com regras de comportamento e interação, tendo o objetivo de criar um comportamento coletivo. Nos anos 1970, um AC de duas dimensões e dois estados foi inventado por John Conway (GARDNER, 1970), chamado de *jogo da vida*, ganhando bastante notoriedade pela comunidade científica. Este jogo reproduz, através de regras simples, as alterações e mudanças em grupos de seres vivos, tendo aplicações em diversas áreas. Mais tarde Wolfram (WOLFRAM, 1994; WOLFRAM, 2002), destacou os ACs como ferramentas para modelar sistemas complexos através de regras simples entre a célula e sua vizinhança. Além disso Toffoli (TOFFOLI, 1984) cita que os sistemas complexos modelados a partir das equações diferenciais também podem ser modelados através de ACs, sendo as equações matemáticas substituídas por regras computacionais.

Na literatura há alguns modelos feitos com o objetivo de se estudar a dinâmica de pedestres. Dentre eles podemos destacar os autômatos celulares (SCHADSCHNEIDER, 2002), os modelos de forças sociais (JIANG et al., 2010), teoria de

filas (CRUZ; MACGREGOR SMITH; MEDEIROS, 2005; CRUZ; OLIVEIRA; DUCZMAL, 2010) e os modelos multi-agentes (FERBER, 1996).

Este trabalho se baseará na teoria de AC para estudos de dinâmica de pedestres. Estudos deste tipo tem chamado a atenção da comunidade por diversas razões. Diversos acontecimentos reais, tais como catástrofes ocorridas por incêndios e concentração de multidões mostram a grande importância de se desenvolver pesquisas acerca do tema e a grande contribuição que essas pesquisas podem oferecer. O estudo da segurança contra tais fenômenos, apesar de ser considerada fundamental no desempenho de projetos de edificações, até pouco tempo era pouquíssimo contemplada devido a complexidade e carência de pesquisas, em parte, devido a limitações computacionais do passado. Hoje em dia esse se tornou um fértil campo de pesquisa.

Além dos estudos em projetos de edificações novos, onde o executor tem grande liberdade e poder de intervenção no espaço a ser criado, hoje, existe um segundo grande campo de atuação, que é a intervenção em espaços existentes. Principalmente nos grandes centros urbanos, identificam-se construções com grande potencial de reuso, mas que necessitam ser adaptadas para se adequarem às condições modernas de relações socio-culturais, de trabalho e moradia. A questão da segurança contra incêndio não pode ser desprezada nessas adaptações, pois uma alteração de uso/ocupação pode aumentar ou diminuir o risco de incêndio e ter consequências desastrosas. Ressalta-se que as medidas de segurança contra incêndio introduzem um custo adicional à edificação, tanto em sua construção como durante o seu uso. Contudo, incorporando essas medidas já na concepção do projeto, é possível torná-las mais efetivas a um menor custo.

Uma das grandes vantagens do uso de AC nos estudos de evacuação é a possi-

bilidade de se avaliar a segurança de um ambiente já existente ou até mesmo de um ambiente a ser construído, além de verificar se o mesmo atende (ou atenderá) certas normas de segurança, apontando caminhos para que essas normas sejam atendidas. Outra grande vantagem da ferramenta estudada neste artigo é a sua grande flexibilidade, ou seja, a possibilidade de se avaliar praticamente qualquer ambiente real e qualquer mudança que possa ser realizada em sua configuração. Por fim, simulações de dinâmica de pedestres permitem a observação de uma ampla quantidade de características ligadas ao fluxo de pessoas e um maior entendimento de seus princípios básicos, tais como suas propriedades emergentes.

Os estudos de dinâmica de pedestres a partir de ACs é uma teoria recente, em que suas principais contribuições aconteceram principalmente nas últimas décadas. O primeiro trabalho a apresentar esta teoria, no formato como será visto neste trabalho, foi desenvolvido por Schadschneider (2002). Este trabalho foi fundamental para apresentar à comunidade científica a teoria de autômatos celulares para dinâmica de pedestres num formato que inclui os fenômenos coletivos mais evidentes em dinâmica de pedestres: a interação entre pedestres e a interação com o ambiente, através da componente chamada de "probabilidade de transição". Vários outros modelos surgiram posteriormente, seguindo e mantendo a mesma essência, porém incluindo novos fenômenos coletivos. Posteriormente, o mesmo autor publicou outros artigos que apresentam novas versões para a chamada probabilidade de transição, ou seja, outras formas de representar os fenômenos coletivos (BURSTEDDE et al., 2001; SCHADSCHNEIDER et al., 2009).

Em 2012, surge mais um trabalho incluindo uma nova interação entre pedestres (SUMA; YANAGISAWA; NISHINARI, 2012), o chamado *campo de antecipação*. Neste tra-

balho os pedestres durante a simulação são capazes de evitar colisões analisando o comportamento dos outros. Em 2013 Pereira *et al.* (PEREIRA; DUCZMAL; CRUZ, 2013) apresentaram um novo trabalho que contém a possibilidade de mudança de velocidade dos pedestres com base na lotação do ambiente, além de apresentar simulações em ambientes complexos, com várias salas e corredores.

Alguns destes trabalhos tem apresentado modelos em que pedestres seguem rotas contrárias e as propriedades emergentes deste comportamento. Em casos de simulação emergencial, este tipo de modelo tem grande importância, pois a evacuação pode ocorrer em mais de uma saída. Porém existem alguns comportamentos que ainda não foram reproduzidos nos modelos anteriores. Um deles é a escolha inicial da rota de saída. Como cada pedestre irá escolher a melhor? A outra é a possibilidade de mudança de rota. Um pedestre pode escolher inicialmente uma rota e com o passar do tempo, perceber que aquela não foi uma boa alternativa, devido a fatores como congestionamentos, por exemplo, decidindo retornar e seguir por uma nova rota. Acredita-se que pela não adição deste comportamento ao modelo pode haver um desvio nos parâmetros de segurança.

Em ocorrências reais de evacuação, apesar da maioria das pessoas buscarem o exterior de forma isolada, podem ocorrer formação de grupos afins, formados por pessoas que caminham ao exterior juntas. Ao invés de uma única pessoa do grupo optar por uma rota alternativa, todos do grupo terão este mesmo comportamento. Estes grupos afins podem ser familiares, amigos ou pessoas que, mesmo sem se conhecerem, resolvem se juntar durante o momento de emergência. Acredita-se que a presença considerável destes grupos podem afetar os parâmetros de segurança, e por isso surge a necessidade

de sua adição ao comportamento simulado.

Neste trabalho estes são os principais comportamentos a serem reproduzidos, através de um novo conjunto de equações e regras que definem a probabilidade de transição.

Este trabalho está dividido da seguinte forma. O Capítulo 2 apresenta inicialmente os principais modelos de sistemas complexos utilizados para estudos de dinâmica de pedestres. Em seguida são apresentados o conceito formal e informal de ACs segundo Wolfram (2002) e suas principais componentes. Neste capítulo também é apresentado os conceitos de ACs determinísticos e estocásticos, que são conceitos importantes para a continuidade do trabalho. Apresenta-se posteriormente os ACs Estocásticos com enfoque na simulação de pedestres. É apresentado também as componentes importantes desse modelo, que são as escalas de célula e velocidade, a matriz de preferências, os campos dinâmico, estático e de antecipação, as propriedades de decaimento e difusão e a probabilidade de transição. Este último será apresentado em várias versões diferentes, desde o primeiro modelo apresentado até o trabalho mais recente.

O Capítulo 3 apresenta o novo modelo de ACs para dinâmica de pedestres com mudança de rotas e campos de grupos. Primeiramente é apresentado a nova versão da probabilidade de transição e em seguida, as novas equações que definem as probabilidades de mudança de rota e definição da rota inicial. Também é apresentada a probabilidade de transição para as células em grupo.

O Capítulo 4 primeiramente apresenta toda a metodologia e ferramentas utilizadas para implementação do modelo. Em seguida são apresentados os resultados das primeiras simulações, feitos de forma subjetiva para uma avaliação prévia e primeiras conclusões. Em seguida são apresentados os resultados das segunda etapa de simulações, cujo

objetivo é realizar uma comparação direta entre o modelo tradicional e o modelo proposto neste trabalho. E por fim a terceira etapa de simulações, que tem como objetivo uma avaliação geral sobre a influência dos novos parâmetros do modelo nas métricas de segurança.

O Capítulo 5 apresenta as principais conclusões, comentários e propostas de trabalhos futuros.

2 FUNDAMENTOS TEÓRICOS

Na literatura existem alguns modelos para descrever as principais características do fluxo de Pedestres. Dentre estes, destacamos os modelos de forças sociais, modelos baseados em teoria de filas, modelos de autômatos celulares e modelos multiagentes. Faremos uma breve revisão sobre estes modelos com uma ênfase maior para os modelos de autômatos celulares, que representam o modelo escolhido para os estudos deste trabalho.

Devido a grande complexidade de estudos de dinâmica de pedestres, os modelos possuem focos em áreas específicas, apesar de ser possível utilizá-los em áreas que não representam o seu foco principal. As áreas de estudo podem ser, por exemplo, passeata de pedestres, movimento de pedestres em cruzamentos, evacuação emergencial, evacuação comum, movimentação de pedestres em um sistema bancário, dentre outros. A seguir é apresentada uma revisão destes métodos de forma resumida, apenas para compreendermos a ideia geral. As informações detalhadas de cada um deles poderão ser obtidas nas referências em destaque. Lembramos que as teorias acerca de dinâmica de pedestres apresentadas a seguir não são únicas. Existem outras teorias recentes, que não serão detalhadas aqui por não ser este o enfoque do trabalho. Para aqueles que desejam conhecê-las, destacamos os trabalhos de Jiang et al. (2010), Li, Kuang e Fan (2012),

Smith et al. (2009), Tian et al. (2009) e Zheng e Liu (2010). Existem algumas que são derivadas dos modelos tradicionais como os modelos hidrológicos (RINALDI et al., 2007) e os modelos *multi-grid* (SONG et al., 2006).

2.1 Modelos de Forças Sociais

No modelo de forças sociais (HELBING; MOLNÁR, 1995) o pedestre é visto como uma partícula sujeita a forças sociais e físicas que são responsáveis pela sua movimentação, através da aceleração derivada da soma de tais forças. Logo, um estímulo sensorial causa uma reação comportamental que depende dos objetivos pessoais e é escolhida dentre um grupo de alternativas que visam maximizar a utilidade do caminho até o destino final.

Neste sentido, podemos dizer que a movimentação de pedestres pode ser descrita como sendo governada por uma força social F_{soc} que representa o efeito do ambiente, como a interação com outros pedestres e obstáculos sobre o comportamento do pedestre k , ($k = 1, \dots, N_p$), sendo N_p o número de pedestres.

A força F_{soc} determina a mudança temporal da velocidade atual $v^k = \frac{dr^k}{dt}$ de um pedestre k , sendo r^k sua posição no espaço no tempo t . Dessa forma temos que a dinâmica de evolução da velocidade no tempo será dado por:

$$m^k \frac{dv^k}{dt} = F_{soc} + F_{atrit}, \quad (2.1)$$

em que m^k é a massa corporal do pedestre k e F_{atrit} o atrito de movimentação. De forma resumida, o modelo de forças sociais para dinâmica de pedestres fornece a interpreta-

ção de que as forças são uma mistura de fatores sociais e físicos. Matematicamente, a força social é a soma de diferentes vetores de força, que representam diferentes estímulos afetando simultaneamente o comportamento do pedestre, como pode ser descrito pela Equação 2.2, que apresenta a força de desejo F_D , a força psicológica F_P e a força granular F_G .

$$F_{soc} = +F_D + F_P + F_G \quad (2.2)$$

A F_D representa o desejo de um pedestre em alcançar o seu objetivo, da forma mais natural. Indica onde está o alvo e a velocidade desejada. A F_P é uma força de repulsão, que indica a intenção de um pedestre em manter certa distância de outros pedestres e de retenções ou paredes. A F_G representa as forças físicas de contato. Essa força entra em ação quando os pedestres ficam tão próximos uns dos outros que ocorre o contato físico. Assim aparece a força de corpo, atuando de forma contrária à compressão e a força de atrito de deslizamento, repelindo o corpo.

Como se pode ver, esta teoria é recente e já apresenta importantes trabalhos, que apresentam variações deste modelo e resultados empíricos a partir de simulações (PARISI; DORSO, 2007; PELECHANO; ALLBECK; BADLER, 2007; SEYFRIED; STEFFEN; LIPPERT, 2006).

2.2 Modelos Baseados em Teoria de Filas

São modelos usados para demonstrar previamente o comportamento de um sistema que ofereça serviços cuja demanda cresce aleatoriamente, tornando possível

dimensioná-lo de forma a satisfazer os clientes e ser viável economicamente para o provedor do serviço, evitando desperdícios e gargalos. Para o modelo, supõe-se que o sistema sempre terá formação de filas, ou seja, a procura por um determinado serviço é sempre maior que a capacidade do sistema. Dessa forma haverá o acúmulo de pedestres. Assim para se organizar melhor a metodologia, o processo é dividido pelos seguintes componentes:

- Processo de chegada (A): indica qual será o padrão de chegada dos clientes. Como o tempo de chegadas acontecem aleatoriamente, utilizam-se distribuições de probabilidade para modelar esse processo, sendo a mais comum a distribuição de Poisson, supondo-se que os tempos entre chegadas são exponencialmente distribuídos.
- Distribuição do tempo de serviço (S): indica qual distribuição de probabilidade será utilizada para modelar o tempo de serviço. Além disso existem as variantes para este componente como a dependência da fila, ou seja, o tempo de serviço pode ser dependente do número de clientes na fila, sendo por exemplo, em média menor em casos de filas longas.
- número de servidores (m): indica a quantidade de pontos no sistema farão o atendimento paralelamente aos clientes, sendo que podemos ter única fila (uma fila para todos os servidores) ou múltiplas filas (uma para cada servidor).
- Capacidade do sistema (K): representa a capacidade máxima de clientes no sistema, que pode ser finita ou infinita. Nos casos de capacidade finita haverá perda de clientes caso chegue algum cliente e a capacidade esteja no limite.

- População (N): indica o tamanho da população que poderá usar o serviço do sistema, que também pode ser finita ou infinita.
- Disciplina (Q): apresenta o modo como os clientes serão atendidos, sendo que o padrão é o sistema *FCFS* (First Come, First Served), que significa que o primeiro a chegar será o primeiro a ser atendido. Padrões como *LCFS* (Last Come, First Served) indicam que o último a chegar será o primeiro a ser atendido, padrão observado, por exemplo, em pilhas de provas a serem corrigidas.

A notação para um sistema de filas segue o padrão $A/S/m/K/N/Q$. Por exemplo podemos ter um sistema $M/M/2/100/5000/FCFS$ que indica um sistema com processo de chegada e de serviço Markoviano (Poisson), dois servidores, capacidade para cem clientes em uma população de cinco mil, num sistema *FCFS*. Em alguns casos podemos ter também uma notação simplificada, como um sistema $M/M/1$ em que assume-se o processo de chegada e de serviço Markoviano, um servidor, capacidade e população infinita, em um processo *FCFS* (valores default).

Dessa forma podemos utilizar várias métricas, como o número de chegadas, tempo de ocupação, taxa de chegada, utilização, tempo médio de serviço, dentre outras, com o objetivo de otimização do sistema, como a alocação ótima da capacidade.

Mais informações podem ser obtidas em alguns dos principais trabalhos que envolvem a teoria. Dentre eles destacamos os trabalhos de Cruz, MacGregor Smith e Medeiros (2005), Cruz, Oliveira e Duczmal (2010) e Woensel e Cruz (2009). A evacuação emergencial não é uma parte da teoria que costuma ser estudada por estes modelos, porém Stepanov e MacGregor Smith (2009) apresentam um caso em que os sistemas de filas são utilizados para designar *layouts* e

rotas ótimas de evacuação.

2.3 Modelos de Sistemas Multiagentes

Os modelos de sistemas multiagentes são modelos compostos por um número de agentes (células) heterogêneos, ou seja, os agentes são modelados para se comportarem de forma autônoma, num sistema baseado a partir de regras locais de interação entre os agentes e o ambiente. Os modelos de simulação multiagentes são utilizados em situações onde a variação individual entre os agentes não pode ser desprezada. Este tipo de simulação permite a modelagem de uma população heterogênea, onde cada agente pode ter motivações e incentivos pessoais. FERBER (1996, 1999) define que o modelo de simulação multiagente está baseado na ideia do relacionamento de um indivíduo com um programa, sendo possível simular um mundo artificial formado por entidades computacionais interativas. Um agente equivale a uma entidade do mundo real, ou um grupo delas. Tais atores podem ser de diferentes naturezas e com granularidades variadas, como, por exemplo, seres humanos, robôs, programas computacionais, objetos inanimados e organizações. Além disso, não existe o estabelecimento de autoridade central, já que os agentes são modelados para comportarem-se de forma autônoma, em um modelo auto-organizável baseado em regras de interação entre os agentes e o ambiente. Cada agente contém estados internos e regras de comportamento, permitindo que interajam com outros agentes e com o ambiente a sua volta. Agentes são também autônomos e demonstram certo grau de iniciativa, permitindo que comportem-se como entidades orientadas a objetivos. Eles são modelados para executar a vasta maioria de suas ações sem nenhuma interferência direta de humanos ou outros agentes computa-

cionais. Exemplos de agentes incluem pessoas, grupos, organizações, insetos sociais, robôs, dentre outros. Por causa dessa amplitude no conceito de modelos multiagentes, muitos autores acredita-se que os sistemas multiagentes são uma nova forma de ciência. Podemos dizer que os modelos de ACs, que serão estudados de forma detalhada a seguir, são casos particulares de modelos multiagentes, onde a parametrização é a mesma para todos os agentes. Para transformá-los em um modelo multiagente mais próximo do conceito dado anteriormente, poderíamos por exemplo, ao invés de fixar os valores dos parâmetros, fornecer uma distribuição de probabilidade para os mesmos, em que cada agente (neste caso particular uma célula) teria valores exclusivos, e por conseguinte, teriam comportamentos e características únicas, apesar delas serem criadas a partir de uma lei que vale para todos. Mais informações poderão ser obtidas em Hamagami e Hirata (2003).

2.4 Autômatos Celulares

2.4.1 Conceitos Iniciais

De uma maneira geral, um autômato celular (AC) pode ser definido como um objeto (a princípio abstrato) composto por duas componentes intrinsecamente ligadas. A primeira é um reticulado regular, discreto (finito ou infinito), que constitui a estrutura subjacente do AC. A segunda é um único autômato finito que se repete de maneira idêntica em cada nó do reticulado. Cada um desses autômatos determinará uma posição do reticulado. Uma determinada posição mantém um intercâmbio com um conjunto finito de posições do mesmo reticulado. Este conjunto denominamos de vizinhança. Esse intercâmbio determina a evolução global do sistema ao longo de passos em tempo

discreto. Veremos agora a definição formal (WOLFRAM, 2002; ILACHINSKI, 2001).

Um automato celular d -dimensional \mathcal{A} é um quarteto $(\mathbb{Z}^d, S, N, \delta)$, onde:

- \mathbb{Z}^d representa o espaço com d dimensões de números inteiros; cada ponto desse espaço, também chamado de reticulado, está ocupado por uma célula;
- S é um conjunto finito, cujos elementos representam os estados possíveis de cada célula de \mathcal{A} ;
- N é um subconjunto finito de \mathbb{Z}^d , chamado vizinhança de \mathcal{A} , onde cada elemento $n \in N$ deve ser interpretado como as coordenadas relativas das células vizinhas, ou seja, dada uma célula c localizada em $p \in \mathbb{Z}^d$, a sua vizinhança $N(c)$ é o conjunto de células localizadas em $p + n$; por simplicidade, iremos supor que $(0, \dots, 0) \in N$;
- $\delta : S^{k+1} \rightarrow S$, é a regra local de transição de \mathcal{A} , onde k é a quantidade de células em N ;

O conjunto S pode possuir estados especiais s , chamados inativos "quiescent" para os quais se verifica que $\delta(s, \dots, s) = s$.

2.4.2 Configuração ou Estado Global de \mathcal{A}

Dado um AC $\mathcal{A} = (\mathbb{Z}^d, S, N, \delta)$ podemos lhe associar uma aplicação $c_t^{\mathcal{A}} : \mathbb{Z}^d \mapsto S$ que fornece para cada instante $t \geq 0$ o estado de todas e cada uma das células que compõem \mathcal{A} . Essa aplicação é conhecida como configuração ou estado global de \mathcal{A} no instante t .

2.4.3 Vizinhanças

Um conceito importante na teoria de AC é o conceito de vizinhança, já que o seu estado atual determina a configuração global no instante posterior. Daremos as definições formais de vizinhança de *von Neumann* e de *Moore*.

Seja c uma célula no ponto z de \mathbb{Z}^d , com $z = (z_1, \dots, z_d)$; consideremos as normas $\|v\|_1 = \sum_{i=1}^d |z_i|$, e $\|v\|_\infty = \max\{|z_i| \mid i \in \{1, \dots, d\}\}$. Então:

- defina $N_V^n(c) = \{x \mid x \in \mathbb{Z}^d, \|z - x\|_1 \leq n\}$. Logo $N_V(c)$ é a vizinhança de *von Neumann* de c com n passos,
- defina $N_M^n(c) = \{x \mid x \in \mathbb{Z}^d, \|z - x\|_\infty \leq n\}$. Logo $N_M(c)$ é a vizinhança de *Moore* de c com n passos.

Neste trabalho utilizaremos apenas as vizinhanças para $n = 1$ ou $n = 2$, ou seja, a vizinhança com um ou dois passos de distância. Como exemplo, apresentamos na Figura 1 as vizinhanças $N_V^n(c)$ e $N_M^n(c)$ (representadas pelas células tracejadas) para $n = 1$ em um espaço \mathbb{Z}^2 .

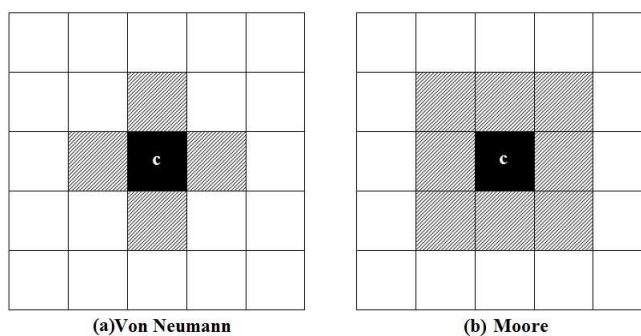


Figura 1: Vizinhanças de *von Neumann* e *Moore*.

2.4.4 Autômatos Celulares Determinísticos e Estocásticos

Como vimos antes, um dos componentes do AC é a regra local de transição. Esta regra determina o estado de uma célula no tempo $t + 1$ a partir do estado das células que compõem a sua vizinhança no tempo t . Como exemplo, vamos supor um AC binário unidimensional com vizinhança um passo. A Tabela 1 apresenta os estados possíveis para uma célula c no tempo $t + 1$ a partir das configurações possíveis de sua vizinhança no tempo t . Note que a sequência final no novo estado é o número binário 00011110, que em decimal equivale ao número 30, logo Wolfram (2002) batizou essa regra como a regra 30. A Figura 2 apresenta a evolução gráfica do autômato dado como exemplo, em que as células de estado 1 são apresentadas por pontos escuros e as células de estado 0 são apresentadas por pontos em branco. Wolfram propôs usar a sua coluna central como um gerador de números pseudo-aleatórios, que, de fato, é apontada como aleatória em muitos testes de aleatoriedade.

O exemplo apresentado é um exemplo típico de um *autômato celular determinístico*, pois segundo a regra apresentada, a vizinhança define de forma única o estado futuro da célula, pois para cada configuração da vizinhança temos um, e apenas um, estado posterior para a célula. Este tipo de AC em especial, possui um padrão final, que é o resultado das interações determinísticas entre a célula e sua vizinhança, e que em muitos casos, podem ser usados como representações de processos naturais. A Figura 2 apresenta o padrão final do autômato celular do exemplo, que se assemelha com o padrão observado em conchas da espécie *Conus Textile*, conforme a Figura 3.

Quando temos mais de um estado para a célula a partir da mesma configuração de sua vizinhança, temos o chamado *autômato celular não determinístico*. Neste caso

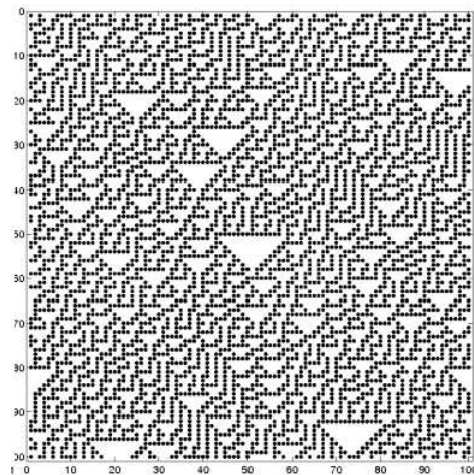


Figura 2: Progressão geométrica de um AC unidimensional de regra 30.



Figura 3: Concha da espécie *Conus Textile* com padrão semelhante à regra 30.

precisamos de mais um componente para definir o estado futuro, sendo o primeiro evidentemente a vizinhança e um segundo componente que pode ser, por exemplo, um fator externo ao modelo. Quando este segundo fator define que a escolha será aleatoriamente, temos o chamado *autômato celular estocástico*. Este último caso é a base para o estudo de dinâmica de pedestres. Na Tabela 2 temos um exemplo de AC estocástico, que representa uma extensão do exemplo da Tabela 1. Neste exemplo temos duas possibilidades para cada célula, sendo que a vizinhança define as probabilidades da célula assumir os valores "0" ou "1", que podem ser dadas por π ou $(1 - \pi)$.

Tabela 1: Exemplo de AC determinístico.

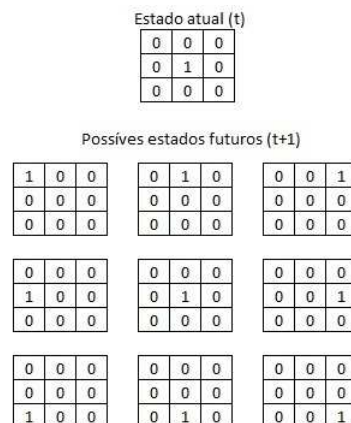
Padrão atual (t)	111	110	101	100	011	010	001	000
Novo estado (t+1)	0	0	0	1	1	1	1	0

Tabela 2: Exemplo de AC estocástico.

Padrão atual (t)	111	110	101	100	011	010	001	000
estado (t+1) Prob. π	0	0	0	1	1	1	1	0
estado (t+1) Prob. $(1-\pi)$	1	1	1	0	0	0	0	1

2.5 Autômatos Celulares Estocásticos para Simulação de Pedestres

Um conhecido modelo de AC estocástico para simulação de pedestre foi desenvolvido por Schadschneider (2002). Este modelo consiste em um AC bidimensional, que apresenta, em sua forma básica, apenas dois estados e vizinhança com um ou dois passos de distância. Para explicar melhor esse modelo, a Figura 4 apresenta uma parte do autômato e seus dois possíveis estados: estado "1" que chamaremos de estado "ocupado" e estado "0" que chamaremos de estado "desocupado". Neste pedaço do autômato está representada uma única célula em estado ocupado e sua vizinhança um passo.

Figura 4: Estados no tempo t e tempo $t + 1$ do modelo de Schadschneider (2002).

Note que, temos nove possíveis estados posteriores, sendo que todos eles possuem apenas uma célula ocupada. Um dos estados possíveis representa uma repetição do estado atual. A partir desse exemplo, podemos interpretar a mudança de estado como um deslocamento da célula ocupada para uma das suas vizinhanças, ou até mesmo a permanência no mesmo ponto do reticulado, conforme resumido na Figura 5. Este deslocamento representa o movimento de uma pessoa em um ambiente através de uma simulação discreta. Evidentemente em casos onde temos uma ou mais vizinhanças ocupadas, esse deslocamento fica limitado e portanto só será realizado para uma vizinhança desocupada. Se todas as vizinhanças estiverem ocupadas, a célula obrigatoriamente ficará na mesma posição. Na Figura 6 temos a fotografia de um ambiente real em situação de evacuação emergencial e sua possível representação através de um AC.

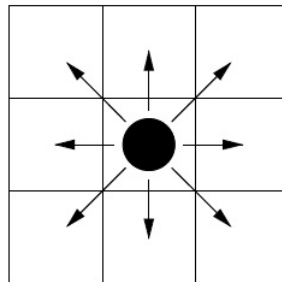


Figura 5: Representação de deslocamento para a vizinhança.

Diferentemente dos modelos clássicos de AC, este modelo permite regras diferentes para grupos de células pertencentes ao mesmo autômato. Em ambientes reais, o mesmo pode ser ocupado por outros objetos além dos pedestres, tais como mesas, cadeiras e qualquer outro objeto que porventura possa existir. Estes objetos são fixos e serão representados no AC por um grupo especial de células, de forma que, este grupo de células só terão um único estado durante toda a simulação: o estado ocupado. Inclui-se

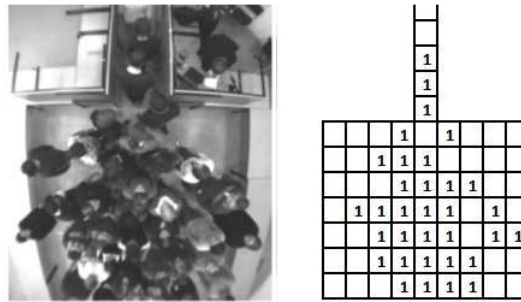


Figura 6: Ambiente real e sua representação por um AC.

neste grupo as células que representam as divisões do ambiente (paredes, muros e portas fechadas).

Este modelo é extremamente eficiente e permite a simulação de grandes multidões, reproduzindo características coletivas e de auto-organização, tais como formação de filas e fluxo de corredores. Para que esse autômato possa realmente representar o comportamento coletivo e que possa reproduzir muitas características de dinâmicas de pedestres, o mesmo possui alguns conceitos e regras importantes, que serão apresentadas ao longo desta seção.

2.5.1 Escala de Células e velocidade

Segundo estudos anteriores (TREGENZA, 1976), a velocidade média de deslocamento de um pedestre em meio a uma multidão é estimada em 1,3 metros por segundos e a área média de ocupação de um pedestre é estimada em $40 \times 40 \text{ cm}^2$. Dessa forma é suposto que cada reticulado do AC terá uma área de $40 \times 40 \text{ cm}^2$, de forma que, este só possa ser ocupado por um pedestre e só assumira dois estados possíveis: ocupada ou vazia. Dessa forma, a cada passo discreto ou mudança de estado, o pedestre se moverá

por 40 centímetros. Com base nestes dados podemos estimar a velocidade média do pedestre em 0,3 m/s a cada passo discreto. Por questão de simplicidade, o grupo de células fixas (mesas, cadeiras o outros) também terão dimensão $40 \times 40 \text{ cm}^2$, mesmo que os objetos reais não ocupem esse valor exato de área. Para grandes objetos, a área será representada por células fixas múltiplas, ou seja, por um número de células que tenham dimensão mais aproximada do objeto real. Logo, com base nestes dados, o tempo de evacuação total será estimado pela seguinte expressão:

$$T = \frac{N_{sim} \times 0,298}{60} \quad (2.3)$$

em que N_{sim} é o numero de passos discretos até a completa evacuação do ambiente.

2.5.2 Matriz de Preferências

Como vimos, neste modelo de AC temos vários estados futuros em que apenas um deles ocorrerá e por isso precisamos associar a este evento uma distribuição de probabilidade. Esta associação será feita através da *matriz de preferências*, que consiste simplesmente em uma matriz de dimensão 3×3 contendo as probabilidades iniciais de deslocamento da célula para cada respectiva posição na vizinhança. Essa associação é feita conforme apresentado na Figura 7. Neste caso temos uma autômato com vizinhança um passo. Nos casos em que temos vizinhança dois passos, a matriz de preferências passa a ter dimensão 5×5 . Deve-se lembrar que estas probabilidades são apenas as orientações iniciais do autômato, e que a probabilidades finais serão definidas segundo este e outros fatores, que serão vistos posteriormente.

A distribuição de probabilidade associada a cada passo pode ser constante ao

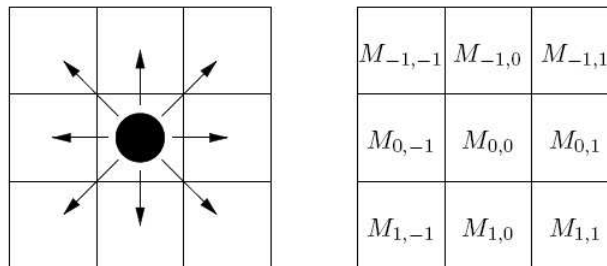


Figura 7: Possíveis deslocamentos e a matriz de preferências $M = (M_{ij})$.

longo de toda a simulação ou sofrer alterações ao longo do processo. No modelo de Schadschneider (2002) assume-se que a matriz de preferências não sofre alterações ao longo da simulação. Em modelos mais avançados como o modelo proposto por Pereira, Duczmal e Cruz (2013) assume-se que a matriz de preferências pode ser uma matriz de dimensão 3×3 ou 5×5 , dependendo da posição que a célula ocupa no ambiente, como veremos adiante. Além disso assume-se que a distribuição de probabilidade atribuída à matriz de preferências é sempre a mesma para todas as células.

Um exemplo simples de matriz de preferências é quando temos um movimento retilíneo, sem flutuações na direção. Neste caso um dos elementos da matriz de preferências carrega toda a massa de probabilidade e todos os outros elementos assumem valor zero de probabilidade. Outro exemplo simples é quando temos um "passeio aleatório" em que todos os elementos da matriz de preferências assumem o mesmo valor. A construção da matriz de probabilidades é um processo que deve levar em conta o conhecimento acerca do fenômeno e/ou estimada através de dados. Entretanto a dificuldade de se conseguir dados para esse fim torna a segunda opção praticamente inviável. O conhecimento acerca do fenômeno aliado aos resultados visualizados durante a simulação comparados aos eventos reais são geralmente as ferramentas para construção

da matriz de preferências e definição de outros parâmetros do modelo. O trabalho de Suma, Yanagisawa e Nishinari (2012) apresenta uma validação do modelo a partir de uma simulação física, feita com pedestres reais e gravada para fins de comparação. Neste trabalho, as comparações entre o modelo e os resultados físicos foram feitas de uma forma semi-quantitativa, através das análises gráficas, indicando uma boa escolha dos parâmetros.

2.5.3 Campo Dinâmico (*Dynamic Floor Field*)

Vimos que as probabilidades de transição de uma célula é determinada inicialmente pela matriz de preferências. Entretanto sabemos que a movimentação de um pedestre também está associada, em partes, à interação com outros pedestres. A probabilidade de um pedestre seguir por um determinado caminho pode aumentar se os outros pedestres, em grande quantidade, adotarem este mesmo caminho. Da mesma forma pode diminuir se nenhum ou poucos pedestres o adotarem. Esta é a ideia de *rastro* (ou *trilha virtual*), que é a tendência do pedestre seguir a multidão, principalmente em ambientes em que o conhecimento acerca do mesmo é restrita. Para introduzir este conceito ao modelo foi criado o CD, ou *campo dinâmico* (*Dynamic Floor Field*), que consiste em alterar a distribuição de probabilidade conforme o fluxo de pedestre numa determinada posição. Matematicamente o CD de uma posição é um contador interno, existente para cada posição. O número de pedestres que visitaram a posição determinará seu CD da seguinte forma: a cada célula que passa pela posição (x, y) o CD da mesma é incrementado em uma unidade. Esse aumento pode ser fortalecido (ou enfraquecido) segundo uma constante de acoplamento. A partir de então, o valor do CD será utilizado por outras células como um atrativo para aquela posição, através do cálculo da probabilidade de

transição, como veremos adiante.

2.5.4 Campo Estático (*Static Floor Field*)

Assim como há uma interação entre os pedestres, há também uma interação entre os pedestres e as partes físicas do ambiente, de forma que, existem partes mais atrativas (ou menos atrativas) que outras. Entre as partes mais atrativas podemos citar as portas e saídas de emergências. As partes menos atrativas incluem as proximidades com paredes de lado contrário às saídas. Para representar essa interação foi criado o *campo estático CE (Static Floor Field)*. Diferentemente do CD, o CE permanece o mesmo durante toda a simulação. Este pode ser determinado manualmente pelo pesquisador ou automaticamente, utilizando alguma regra. Schadschneider (2002) utiliza como CE a distância entre a posição da célula e a saída, sendo que, quanto maior a distância, menor será a probabilidade de ocupação da posição. A Figura 8 apresenta um exemplo onde temos o CE para dois ambientes, uma sala simples e um corredor composto por duas salas. Neste caso o CE é representado pelo número de passos até a saída definitiva do ambiente.

Independente de como seja definido o CE, a ideia por trás dele é sempre a mesma: atrair as células para a saída de interesse. Neste trabalho o CE será usado para direcionar a matriz de probabilidades de transição, atribuindo maior probabilidade para as posições com menor valor de CE, portanto, utilizar o CE definido por Schadschneider (2002) ou utilizar o número de passos como CE não terá diferenças significativas.

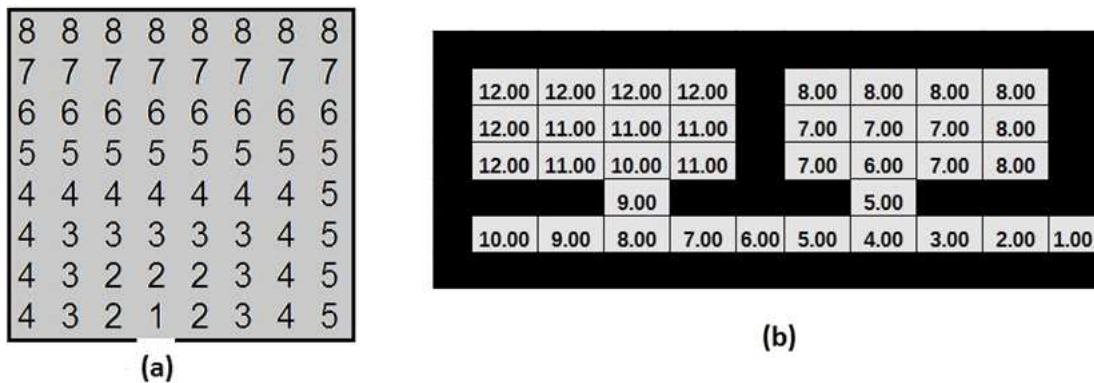


Figura 8: Exemplo de CE definido a partir da contagem de passos.(a) Sala simples e (b) corredor e salas

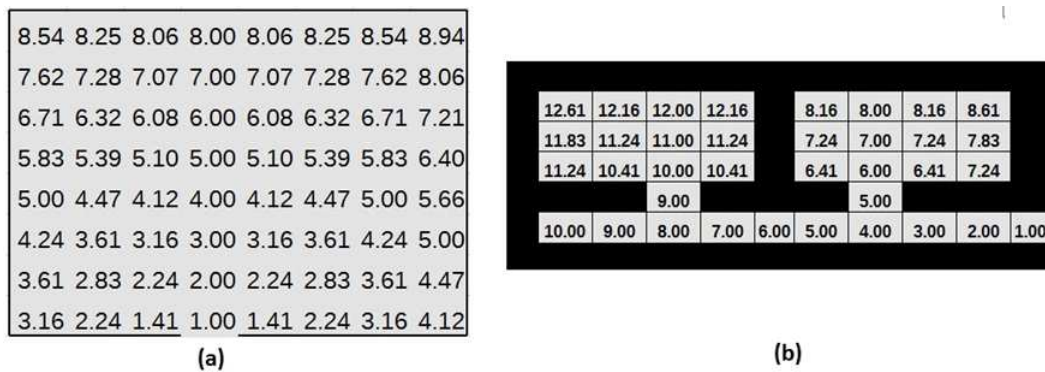


Figura 9: Exemplo de CE definido a partir da distância euclidiana. (a) Sala simples e (b) corredor e salas

2.5.5 Campo de Antecipação (*Anticipation Floor Field*)

Em ambientes onde há mais de um sentido preferencial, existe uma tendência de colisões de pedestres devido ao fluxo inverso ou cruzado. Porém existe também uma tendência dos pedestres em evitar essas colisões antecipando mentalmente o caminho a ser feito por ele e pelos outros pedestres. Isso acaba criando um efeito de desvio temporário de rota, ocasionado por um campo de força imaginário que afasta o pedestre daqueles outros em rota contrária ou cruzada. A esse campo damos o nome de *campo de*

antecipação CA (Anticipation Floor Field) definido por Suma, Yanagisawa e Nishinari (2012). Podemos resumir matematicamente o CA de uma posição de célula como sendo a contagem do número de células que deverão passar por ela em alguns segundos, sendo que, quanto maior o CA, menor é a probabilidade de se dirigir para aquele ponto. Assim, para um célula qualquer, isso diminuirá as probabilidades da mesma se deslocar para aquela posição, mesmo que ela esteja em seu caminho preferencial, determinado pelo CD e CA. Para cada célula deste trabalho é feita uma leitura do CA de uma ou duas posições a sua frente, de forma que, quanto maior o valor de CA, maior é o número de células que pretendem passar pela mesma posição, diminuindo assim suas chances de se deslocar para a mesma posição, criando consequentemente um desvio temporário.

2.5.6 Probabilidades de Transição

Vimos que a matriz de preferências conterá as probabilidades iniciais de transição e que essas probabilidades são modificadas segundo o CD, CE e CA. Essas novas probabilidades são chamadas de *probabilidades de transição*. Este é o principal componente do AC estocástico para simulação de pedestres e tem sido amplamente estudado e modificado, sendo o ponto chave para uma apresentação confiável do modelo, pois é o componente que faz a ligação entre o modelo matemático e todas as propriedades reais da dinâmica de pedestres. Ao longo dos anos, foram criadas várias versões para a equação que determina as probabilidades de transição. Aqui serão apresentadas as versões que mais se aproximam desse trabalho. Schadschneider (2002) apresentada a primeira

versão dessa equação, descrita a seguir:

$$p_{ij} = NM_{ij}D_{ij}S_{ij}(1 - n_{ij}) \quad (2.4)$$

em que:

- p_{ij} é a probabilidade de transição, ou seja, é a probabilidade de uma célula que está em na posição (i^*, j^*) , se deslocar para a posição (i, j) (sendo (i, j) uma vizinhança de (i^*, j^*));
- N é uma constante normalizadora para garantir que $\sum_{i=1}^3 \sum_{j=1}^3 p_{ij} = 1$;
- D_{ij} é o valor da força dinâmica referente a posição (i, j) ;
- S_{ij} é o valor da força estática referente a posição (i, j) ;
- M_{ij} é a probabilidade inicial de transição para a posição (i, j) situada na vizinhança. M_{ij} corresponde a um elemento da matriz de preferências M ;
- n_{ij} é um identificador de estado da posição (i, j) . Se $n_{ij} = 1$ então a posição (i, j) possui estado *ocupado*. Se $n_{ij} = 0$ então a posição (i, j) possui estado *livre*. Deve-se lembrar que os AC estocásticos para simulação de pedestres só possuem dois estados possíveis;

Em Kirchner e Schadschneider (2002) é apresentado um novo modelo de equação para as probabilidades de transição com algumas mudanças importantes. Neste novo modelo vemos pela primeira vez as constantes de acoplamento e o identificador de obstáculos ξ_{ij} . Esta equação não apresenta a matriz M , portanto as probabilidades de transição são calculadas exclusivamente a partir da força dinâmica D_{ij} e da força estática

S_{ij} . Por questões óbvias, os componentes da equação anterior que estão presentes nos modelos posteriores não serão novamente listados. Serão apresentados apenas os novos componentes. O modelo de equação é apresentado a seguir:

$$p_{ij} = Ne^{K_d D_{ij}} e^{K_s S_{ij}} (1 - n_{ij}) \xi_{ij} \quad (2.5)$$

em que

- K_d é a constante de acoplamento que altera os efeitos de D_{ij} ;
- K_s é a constante de acoplamento que altera os efeitos de S_{ij} ;
- ξ_{ij} é um fator reconhecedor de barreiras e obstáculos. Diferentemente dos AC clássicos, Nos AC para simulação de pedestres existem posições que assumem um comportamento especial: possuem o estado *ocupado* durante toda a simulação para representar barreiras físicas. Como a ocupação não se dará por uma célula, o modelo apresenta o identificador particular ξ_{ij} para essas posições. Se $\xi_{ij} = 0$, então a posição (i, j) é uma posição especial, ou seja, é uma barreira física.

Uma nova versão da equação foi apresentada em Suma, Yanagisawa e Nishinari (2012) em que as probabilidades de transição são calculadas levando-se em consideração o CA. A equação apresentada é a seguinte:

$$p_{ij} = Ne^{K_d D_{ij}} e^{K_s S_{ij}} e^{-K_d E_{ij}^{(m)}} (1 - n_{ij}) \xi_{ij} \quad (2.6)$$

$$E_{ij}^{(m)} = \sum_{m' \neq m} A_{ij}^{(m')} \quad (2.7)$$

onde

- m corresponde a direção preferencial da célula, indicada pelo CE;
- m' corresponde às quatro direções que a célula pode seguir em uma vizinhança de *von Neumann*, sendo que 1: direita, 2:cima, 3:esquerda e 4:baixo. Dessa forma $m' \in \{1, 2, 3, 4\}$;
- $A_{ij}^{(m')}$ corresponde ao CA para a posição (i, j) . Logo, cada posição (i, j) existem quatro valores de CA, um para cada direção. Primeiro verifica-se se a posição (i, j) é uma *posição reservada*, ou seja se a mesma faz parte de um conjunto de D_A células pertencentes ao trajeto esperado de algum pedestre (Figura 11). Depois faz se a contagem do número de trajetos esperados que passam por (i, j) e atribui a cada $A_{ij}^{(m')}$ o número de trajetos esperados na direção m' (Figura 10).
- $E_{ij}^{(m)}$ é o efeito da antecipação. Note que, quanto maior o valor de $E_{ij}^{(m)}$, menor será a probabilidade da célula se deslocar para (i, j) . Isso significa que existe um ou mais trajetos esperados de outras células em (i, j) (com diferente CE), logo, deve-se evitar esse trajeto como uma forma de evitar colisões. Mais detalhes podem ser obtidos em Suma, Yanagisawa e Nishinari (2012).
- K_a é a constante de acoplamento que potencializa os efeitos de $E_{ij}^{(m)}$.

Posteriormente, uma nova versão para a equação de probabilidades de transição foi apresentada por Pereira, Duczmal e Cruz (2013). Esta versão contém simultaneamente

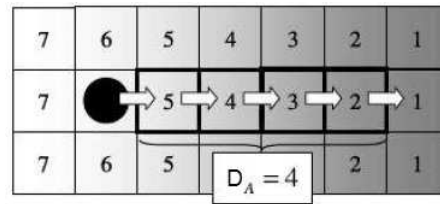


Figura 10: Exemplo de *posições reservadas* para $D_A = 4$ (destacadas pelas linhas sólidas). Os números em cada posição corresponde ao CE (distância da saída). A célula representado pelo círculo sólido tem como trajeto esperado aquele em que há um decréscimo do CE.

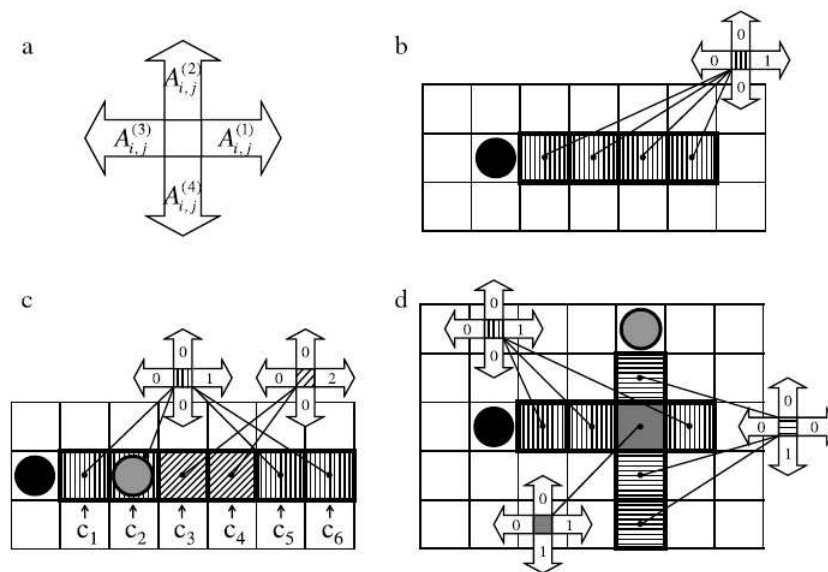


Figura 11: (a) Relação entre $A_{ij}^{(m')}$ e as direções. (b)(c)(d) Exemplos de cálculo de $A_{ij}^{(m')}$ a partir de $D_A = 4$. Os círculos sólidos preto e marrom representam células.

dois tipos de matriz M , as chamadas matrizes $M1$, de dimensão 3×3 e $M2$, de dimensão 5×5 . Logo as probabilidades de transição podem ser atribuídas também para vizinhanças 2 passos, sendo essa uma forma de aumentar a velocidade da célula nos casos em que temos um ambiente com menor densidade ou para pedestres que estejam afastados

da multidão. A equação é apresentada abaixo:

$$p_{ij} = Ne^{K_d D_{ij}} e^{K_s S_{ij}} (1 - n_{ij}) M1_{ij}^{V_{ij}} M2_{ij}^{(1-V_{ij})} \xi_{ij} \quad (2.8)$$

em que

- $M1_{ij}$ é a probabilidade inicial de transição para a posição (i, j) considerando uma vizinhança de 1 passo;
- $M2_{ij}$ é a probabilidade inicial de transição para a posição (i, j) considerando uma vizinhança de 2 passos;
- V_{ij} é uma variável binária, em que $V_{ij} = 1$ se pelo menos uma das duas células na vizinhança de 2 passos no sentido preferencial (de menor distância para a saída) estiverem no estado ocupado. Caso contrário temos que $V_{ij} = 0$.

Este foi o primeiro modelo a apresentar simulações em ambientes complexos, composto por vários ambientes simples (salas ou corredores), em que a saída do ambiente se dá após a migração por vários destes. Neste modelo é usado uma segunda força estática, feita através de um mapeamento do ambiente pelo pesquisador e fornecido ao programa para indicar o sentido preferencial, de forma a reorganizar a matriz de preferências. Neste trabalho é proposto um modelo que utiliza esta mesma técnica de direcionamento, porém de uma forma mais robusta, onde essa reorganização é feita automaticamente utilizando o CE apresentado na Subseção 2.5.4.

2.5.7 Difusão e Decaimento

Vimos que o CD cria uma trilha pelo autômato e essa trilha atrai as células. Porém é lógico pensarmos que as posições vizinhas dessa trilha devam ser mais atrativas que as posições afastadas da mesma. Seria uma forma de levar a célula até a trilha, e mesmo que se desvie, terá uma alta chance de regressar. Em um processo real os pedestres visualizam a rota feita pelos outros pedestres e se deslocam até o mais próximo que conseguirem dela, formando assim uma trilha larga de pedestres, que tende a aumentar com o passar do tempo. Para representar esse comportamento o modelo de AC apresenta um processo chamado de *difusão* em que uma certa quantidade do CD é distribuída para as posições vizinhas.

É consenso também que o CD se enfraquece com o passar do tempo, num processo chamado de *decaimento*. Em estágios avançados, as multidões tendem a consolidar uma rota e qualquer CD fora dessa rota não exerce impacto tão significativo. Dessa forma é preciso diminuir interferência desses CD nas probabilidades de transição. Além disso quando uma trilha recebe poucos ou nenhum pedestre, ela se torna pouco atrativa e se desfaz a longo prazo. O processo de difusão e decaimento está ilustrado na Figura 12. Em Nowak e Schadschneider (2012) é apresentada a equação de difusão e decaimento mais utilizada em modelos de AC para dinâmica de pedestres. A cada passo de simulação, o CD é atualizado de acordo com a seguinte expressão:

$$D_{ij}^{(t+1)} = (1 - \delta)D_{ij}^{(t)} + \beta\Delta D_{ij}^{(t)} \quad (2.9)$$

onde

$$\Delta D_{ij}^{(t)} = D_{i,j+1}^{(t)} + D_{i,j-1}^{(t)} + D_{i+1,j}^{(t)} + D_{i-1,j}^{(t)} - 8D_{ij}^{(t)} \quad (2.10)$$

sendo que

- $D_{ij}^{(t)}$ é o valor de D_{ij} no tempo (t);
- $\beta = \frac{\alpha(1-\delta)}{8} \in (0, 1/8)$ é a constante de difusão;
- $\delta \in (0, 1/2)$ é a constante de decaimento.

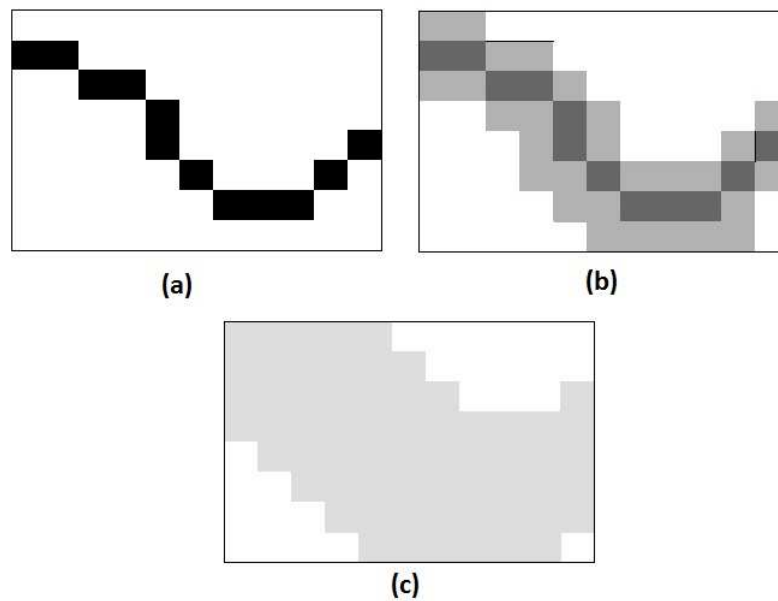


Figura 12: Representação da difusão e decaimento do CD. Em (a) temos um CD recente. Em (b) temos difusão representada pelas células vizinhas em um tom cinza. Em (c) temos a difusão em estágio mais avançado. O decaimento é representado pelo clareamento do CD (note que a cor preta se espalha e perde a tonalidade com o passar do tempo).

2.6 Observações finais

Neste capítulo vimos os principais modelos utilizados para estudos de dinâmica de pedestres e evacuação emergencial. Vimos os conceitos básicos destes modelos, com referência a produções bibliográficas importantes. Em seguida, apresentamos os conceitos básicos de autômatos celulares, mostrando todas as definições importantes para a continuidade deste trabalho. A seguir, apresentamos os conceitos de autômatos celulares voltados para a dinâmica de pedestres, detalhamos as componentes do modelo desenvolvidas até o presente momento e apresentamos por último as diferentes versões da probabilidade de transição, em ordem cronológica. Todas as definições vistas até aqui procuram preparar o leitor para entendimento dos capítulos seguintes, onde apresentaremos de fato, as novas contribuições.

3 AUTÔMATOS CELULARES COM MUDANÇA DE ROTA

Até o momento, os modelos de AC criados apresentaram propostas com o intuito principal de tornar a simulação de dinâmica de pedestres cada vez mais próxima de uma situação real, acrescentando ao mesmo novas componentes que terão a capacidade de simular o comportamento do pedestre com maior eficiência. De fato, em muitos casos os modelos tem conseguido atingir os seus objetivos. A cada nova contribuição, os modelos vão se tornando mais completos e eficientes, podendo ser usados em situações mais complexas e a partir daí, surgem novas necessidades. Por exemplo, com as contribuições de Suma, Yanagisawa e Nishinari (2012) e Nowak e Schadschneider (2012), os modelos de AC já podem ser usados para simular evacuações em ambientes com mais de uma população, onde as rotas de fuga se diferenciam para cada uma delas, e de acordo com Pereira, Duczmal e Cruz (2013) podemos simular ambientes complexos, com várias salas e corredores, tendo até mesmo mais de uma saída. Porém através de estudos feitos a partir desses modelos, surgiram outros problemas a serem analisados. Um deles é a possível mudança de rota de fuga de um pedestre quando a sua rota original se encontra impossibilitada por congestionamento. Supõe-se que em uma situação real o pedestre possa "desistir" de seguir por um caminho obstruído se existir outras opções, principalmente se ainda estiver distante da saída escolhida. Nas simulações realizadas observa-se

que o tempo de evacuação possa estar sendo superestimado devido a não mudança de rota, ao observar uma concentração gigante próximo a uma das saídas enquanto outras estão livres. Neste capítulo apresentaremos um modelo de AC em que a sua principal contribuição é a possibilidade de mudança de rota de um pedestre, além de apresentar mudanças na relação entre a probabilidade de transição e a matriz de preferências. Este modelo chamaremos de *modelo de AC com mudança de rota* sendo a rota estritamente relacionado ao CE. Dessa forma esperamos obter um modelo que possibilite uma auto organização em casos de congestionamentos, com a reformulação de trajetos feita pelos próprios pedestres, de forma a se obter uma simulação mais realista. Para este objetivo, precisa-se definir as situações que propiciarão a mudança de rota. Neste trabalho definimos duas situações críticas. A primeira é a fronteira do congestionamento. Definimos que um pedestre, ao encontrar pela frente um congestionamento, poderá mudar de rota com uma certa probabilidade. A segunda situação crítica é o interior do congestionamento. Suponha que, por um motivo qualquer, um pedestre esteja preso a um congestionamento, porém sua intenção é de seguir uma rota contrária à grande maioria dos que também estão no interior do congestionamento. Neste caso ele pode ficar impossibilitado de continuar o seu percurso e com certa probabilidade, poderá desistir e mudar de rota, acompanhando a maioria. Evidentemente essas probabilidades de mudança de rota devem mudar conforme a posição do pedestre no ambiente. Se o mesmo atinge um congestionamento, porém está muito próximo à sua saída de interesse, então ele terá uma probabilidade reduzida de mudar de rota, se comparado a outros pedestres que estão na mesma situação porém mais distantes da saída de interesse. Logo a distância ou CD será um fator decisivo no cálculo da probabilidades de mudança de rota, de forma que, quanto mais próximo da saída de sua rota, menor será a probabilidade de

mudança de rota. Também será analisada outra situação que pode causar uma mudança de rota: o comportamento de grupo. Em uma situação de evacuação, se o pedestre possuir um grupo interno (grupo de amigos, familiares, estudantes, etc) então pode ocorrer uma mudança de rota em coletivo, mesmo que o pedestre em questão esteja livre para seguir o seu trajeto. para isso basta que um outro elemento do grupo esteja em congestionamento, influenciando o comportamento de todos os outros.

Anteriormente vimos os modelos de probabilidades de transição e sua evolução ao longo do tempo. Será apresentado agora o modelo de probabilidades de transição a ser estudado, que reúne todas as componentes estudadas pelos modelos anteriores. O modelo a seguir é basicamente uma unificação das ideias apresentadas em Pereira, Duczmal e Cruz (2013) e Suma, Yanagisawa e Nishinari (2012). Em seguida as novas componentes para determinação das probabilidades de mudança de rota são apresentadas. Por fim serão apresentados a expressão da probabilidade de transição para grupos e as condições iniciais.

3.1 Probabilidades de Transição

Seja (i^*, j^*) a posição atual da célula e (i, j) uma posição de sua vizinhança $N_M^w(i^*, j^*)$ (lê-se *vizinhança de Moore* de w passos referente à posição (i^*, j^*)). A probabilidade de transição de uma célula da posição (i^*, j^*) para uma posição (i, j) é dada por:

$$p_{ij} = N e^{K_d D_{ij}} e^{-K_a E_{ij}^{(m)}} \left(e^{K_s M_{ij}} \right)^{V_{ij}} \left(e^{K_s U_{ij}} \right)^{(1-V_{ij})} (1 - n_{ij}) \xi_{ij} \quad (3.1)$$

em que M_{ij} é a probabilidade inicial de transição (elemento da matriz de preferências

3x3) da posição (i, j) considerando uma vizinhança de 1 passo e U_{ij} é a probabilidade inicial de transição (elemento da matriz de preferências 5×5) da posição (i, j) considerando uma vizinhança de 2 passos.

Segundo a Figura 5, existem oito possibilidades de deslocamento. Para cada possibilidade de deslocamento será relacionada a uma matriz de preferências, sendo que o elemento $m_1 \in M^{(a)}$ será o maior valor da matriz e ocupará a posição que indica o sentido preferencial, conforme abaixo:

$$M^{(1)} = \begin{bmatrix} a & b & c \\ d & e & m_1 \\ f & g & h \end{bmatrix}; \text{ Sentido preferencial para leste.}$$

$$M^{(2)} = \begin{bmatrix} c & m_1 & h \\ b & e & g \\ a & d & f \end{bmatrix}; \text{ Sentido preferencial para norte.}$$

$$M^{(3)} = \begin{bmatrix} f & d & a \\ g & e & b \\ h & m_1 & c \end{bmatrix}; \text{ Sentido preferencial para sul.}$$

$$M^{(4)} = \begin{bmatrix} h & g & f \\ m_1 & e & d \\ c & b & a \end{bmatrix}; \text{ Sentido preferencial para oeste.}$$

$$M^{(5)} = \begin{bmatrix} b & c & m_1 \\ a & e & h \\ d & f & g \end{bmatrix}; \text{ Sentido preferencial para nordeste.}$$

$$M^{(6)} = \begin{bmatrix} d & a & b \\ f & e & c \\ g & h & m_1 \end{bmatrix}; \text{ Sentido preferencial para sudeste.}$$

$$M^{(7)} = \begin{bmatrix} m_1 & h & g \\ c & e & f \\ b & a & d \end{bmatrix}; \text{ Sentido preferencial para noroeste.}$$

$$M^{(8)} = \begin{bmatrix} g & f & d \\ h & e & a \\ m_1 & c & b \end{bmatrix}; \text{ Sentido preferencial para sudoeste.}$$

Assim, a matriz de preferências escolhida pela célula dependerá exclusivamente do CE, ou seja, da distância (ou número de passos) entre cada vizinhança e a saída de sua rota. A matriz de preferências a ser escolhida é aquela que eleva a probabilidade de transição para a posição da vizinhança com menor CE, atribuindo o valor de m_1 da matriz de preferências para a mesma. Logo, a escolha de M_{ij} , a matriz de preferências efetivamente utilizada pela célula no momento (t) será feita através da seguinte equação:

$$M_{ij} = \prod_{a=1}^8 (M_{ij}^{(a)})^{Q_a} \quad (3.2)$$

sendo $M_{ij}^{(a)}$ o elemento da matriz $M^{(a)}$ ($a = 1, \dots, 8$), que corresponde a posição (i, j) do autômato (o elemento de posição $(2, 2)$ da matriz $M^{(a)}$ corresponde à posição (i^*, j^*) do autômato). A equação 3.2 simplesmente identifica qual $M_{ij}^{(a)}$ será atribuída a M_{ij} através de uma variável binária Q_a estritamente relacionada ao CE da vizinhança um passo da célula (i^*, j^*) .

Definimos como *posição preferencial* da vizinhança, aquela com menor valor de S_{ij} , pois se encontra mais próxima da saída parcial ou definitiva do autômato e portanto deve receber maior valor de probabilidade. Logo os valores de Q_a , afim de selecionar a matriz preferencial correta, são definidos da seguinte forma:

$$Q_1 = 1, Q_a = 0, a \neq 1, \text{ se } S_{(i^*, j^*+1)} = \min(S_{ij} \mid (i, j) \in N_M^1(i^*, j^*));$$

ou seja, Q_1 será igual a um e todos os outros Q_a serão iguais a zero se a vizinhança no sentido leste $(i^*, j^* + 1)$ tiver o menor valor de S_{ij} dentre todos pertencentes à vizinhança 1 passo de (i^*, j^*) . Assim todas as matrizes $M_{ij}^{(a)}$, $a = 2, \dots, 8$ se anulam em (3.2). E assim vale para todos os outros casos:

$$Q_2 = 1, Q_a = 0, a \neq 2, \text{ se } S_{(i^*-1, j^*)} = \min(S_{ij} \mid (i, j) \in N_M^1(i^*, j^*));$$

$$Q_3 = 1, Q_a = 0, a \neq 3, \text{ se } S_{(i^*+1, j^*)} = \min(S_{ij} \mid (i, j) \in N_M^1(i^*, j^*));$$

$$Q_4 = 1, Q_a = 0, a \neq 4, \text{ se } S_{(i^*, j^*-1)} = \min(S_{ij} \mid (i, j) \in N_M^1(i^*, j^*));$$

$$Q_5 = 1, Q_a = 0, a \neq 5, \text{ se } S_{(i^*-1, j^*+1)} = \min(S_{ij} \mid (i, j) \in N_M^1(i^*, j^*));$$

$$Q_6 = 1, Q_a = 0, a \neq 6, \text{ se } S_{(i^*+1, j^*+1)} = \min(S_{ij} \mid (i, j) \in N_M^1(i^*, j^*));$$

$$Q_7 = 1, Q_a = 0, a \neq 7, \text{ se } S_{(i^*-1, j^*-1)} = \min(S_{ij} \mid (i, j) \in N_M^1(i^*, j^*));$$

$$Q_8 = 1, Q_a = 0, a \neq 8, \text{ se } S_{(i^*+1, j^*-1)} = \min(S_{ij} \mid (i, j) \in N_M^1(i^*, j^*)).$$

Os cálculos associados a U_{ij} seguem a mesma metodologia, sendo que U é uma matriz 5×5 , permitindo um deslocamento em uma vizinhança 2 passos. Assim, o calculo

de U_{ij} é definido como:

$$U_{ij} = \prod_{a=1}^8 (U_{ij}^{(a)})^{Q_a}. \quad (3.3)$$

Definiremos $u_1 \in U^{(a)}$ como sendo o ponto de maior probabilidade. Assim, cada matriz $U^{(a)}$ está associada a um sentido preferencial conforme apresentado abaixo:

$$U^{(1)} = \begin{bmatrix} a & b & c & d & e \\ f & g & h & i & j \\ k & l & m & n & u_1 \\ o & p & q & r & s \\ t & u & v & x & z \end{bmatrix}; \text{ Sentido preferencial para leste.}$$

$$U^{(2)} = \begin{bmatrix} e & j & u_1 & s & z \\ d & i & n & r & x \\ c & h & m & q & v \\ b & g & l & p & u \\ a & f & k & o & t \end{bmatrix}; \text{ Sentido preferencial para norte.}$$

$$U^{(3)} = \begin{bmatrix} t & o & k & f & a \\ u & p & l & g & b \\ v & q & m & h & c \\ x & r & n & i & d \\ z & s & u_1 & j & e \end{bmatrix}; \text{ Sentido preferencial para sul.}$$

$$U^{(4)} = \begin{bmatrix} z & x & v & u & t \\ s & r & q & p & o \\ u_1 & n & m & l & k \\ j & i & h & g & f \\ e & d & c & b & a \end{bmatrix}; \text{ Sentido preferencial para oeste.}$$

$$U^{(5)} = \begin{bmatrix} c & d & e & j & u_1 \\ b & h & i & n & s \\ a & g & m & r & z \\ f & l & p & q & x \\ k & o & t & u & v \end{bmatrix}; \text{ Sentido preferencial para nordeste.}$$

$$U^{(6)} = \begin{bmatrix} k & f & a & d & c \\ o & l & g & h & d \\ t & p & m & i & e \\ u & q & r & n & j \\ v & x & z & s & u_1 \end{bmatrix}; \text{ Sentido preferencial para sudeste.}$$

$$U^{(7)} = \begin{bmatrix} u_1 & s & z & x & v \\ j & n & r & q & u \\ e & i & m & p & t \\ d & h & g & l & o \\ c & b & a & f & k \end{bmatrix}; \text{ Sentido preferencial para noroeste.}$$

$$U^{(8)} = \begin{bmatrix} v & u & t & o & k \\ x & n & r & q & f \\ z & i & m & p & a \\ s & h & g & l & b \\ u_1 & j & e & d & c \end{bmatrix}; \text{ Sentido preferencial para sudoeste.}$$

3.2 Regras para Mudança de Direção

Seja \mathcal{A} um autômato celular estocástico para simulação de pedestres bidimensional definido com mais de um CE. Neste caso podemos dizer que, em um contexto de evacuação, temos diversas rotas de saída, diretamente relacionadas aos CE. Inicialmente cada célula ocupada terá um CE definido aleatoriamente, de acordo com a sua posição inicial. Este CE poderá ser alterado durante a simulação quando as vizinhanças n_z passos no seu sentido preferencial estiverem ocupadas, conforme a Figura 13. A vizinhança hachurada chamaremos de *vizinhança preferencial*. Deve-se lembrar que o número de passos que define a vizinhança n_z é usado exclusivamente para determinar a mudança de rota, e que não precisa ser necessariamente igual ao valor n que define a vizinhança para as probabilidades de transição.

Também há algumas regras relativas à parte não hachurada da vizinhança, ou *vizinhança não preferencial*. Podemos supor que, em casos de emergências, geralmente os pedestres que se encontram no início do congestionamento provocado pela multidão poderão mudar de rota, ao passo que, aqueles que estão dentro do aglomerado, mesmo que queiram mudar de rota, não terão como fazê-lo. Assim, uma forma de avaliar se uma célula está ou não incluída em um congestionamento é analisando a ocupação em

sua vizinhança. Se as vizinhanças preferenciais e não preferenciais estiverem todas ocupadas, é um sinal de que essa célula encontra-se dentro de um congestionamento. Se há uma ocupação total na vizinhança preferencial e uma ocupação média ou baixa na vizinhança não preferencial, é um indicativo de que a célula está no início de um congestionamento, podendo assim, se preferir, mudar de rota.

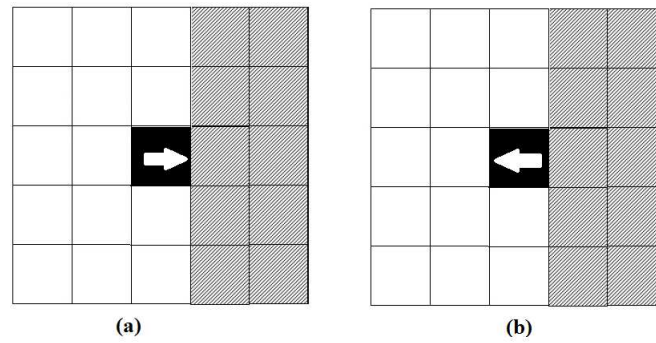


Figura 13: Representação da primeira regra de mudança de rota. As setas indicam o sentido preferencial determinado pelo CE. Em (a) as células hachuradas indicam que estão ocupadas. Em (b), com probabilidade μ , o sentido preferencial é alterado devido a configuração das células ocupadas em uma vizinhança 2 passos ($\varphi=0$).

Seja W o número de CE presentes em \mathcal{A} . Denominaremos como $S_{ij}^{(t)}$ o valor do CE da célula de referência no passo (t). Suponha que, no passo (t), a célula de referência tenha como orientação o CE chamado de $w_g \in w_p$, ($p = 1, 2, \dots, W$) de forma que $S_{ij}^{(t)} = S_{ij}(w_g)$. Suponha agora que, no passo (t), todas as células na vizinhança preferencial n_z passos estão ocupadas e a ocupação na vizinhança não preferencial é menor ou igual a φ (veja Figura 13- indicativo de congestionamento). Logo temos que $P(S_{ij}^{(t+1)} = S_{ij}(w_g)) = \mu_g$, e $P(S_{ij}^{(t+1)} = S_{ij}(w_p)) = \mu_p$, ($\mu_1 + \mu_2 + \dots + \mu_W = 1$).

Evidentemente, podemos dizer que a mudança de rota terá uma maior probabilidade de ocorrer para pedestres que se encontram distantes das saídas e essa probabilidade irá diminuir conforme o pedestre se aproxima da saída escolhida por ele, de forma que,

essa probabilidade se tornará quase nula quando o pedestre se encontrar muito próximo da saída, mesmo que hajam congestionamentos. Logo, para cálculo de $\mu = (\mu_1, \dots, \mu_W)$ devemos ter como base a distância a ser percorrida pela célula até a saída definitiva do ambiente. Seja \mathcal{A} um autômato celular estocástico para simulação de pedestres bidimensional definido com W campos estáticos. Logo definimos

$$q_p = \left(\frac{\frac{1}{s_{ij}(w_p)}}{\sum_{l=1}^W \frac{1}{s_{ij}(w_l)}} \right), p = 1, \dots, W, \quad (3.4)$$

$$\mu_g = q_g^{K_r}, \quad (3.5)$$

$$\mu_p = \left(1 - q_g^{K_r}\right) \frac{q_p}{\left(\sum_{l=1}^W q_l\right) - q_g}, g \neq p, \quad (3.6)$$

em que $\mu_g + \sum \mu_p = 1$. As equações 3.4, 3.5 e 3.6 definem a probabilidade de mudança de rota com base nas distâncias entre o pedestre e as saídas do ambiente, calculadas conforme a Figura 14. Dessa forma, as equações definem que, quanto mais próximo à saída de sua rota, menor será a probabilidade de mudança de rota. Da mesma forma, quanto mais distante, maior será a probabilidade de mudança de rota. A constante K_r , ($0 \leq K_r \leq 1$) é uma constante de acoplamento, que define a força das distâncias no cálculo das probabilidades. Quanto menor o valor de K_r , menor é a força das distâncias, de forma que, para valores baixos de K_r , teremos poucas, ou nenhuma mudança de rota no modelo. Assim, quando $K_r = 0$, o modelo se transforma nos modelos tradicionais. Valores de altos K_r (próximo de 1) indica uma maior força das distâncias, de forma que, teremos um modelo mais propenso a mudança de rota pelos pedestres.

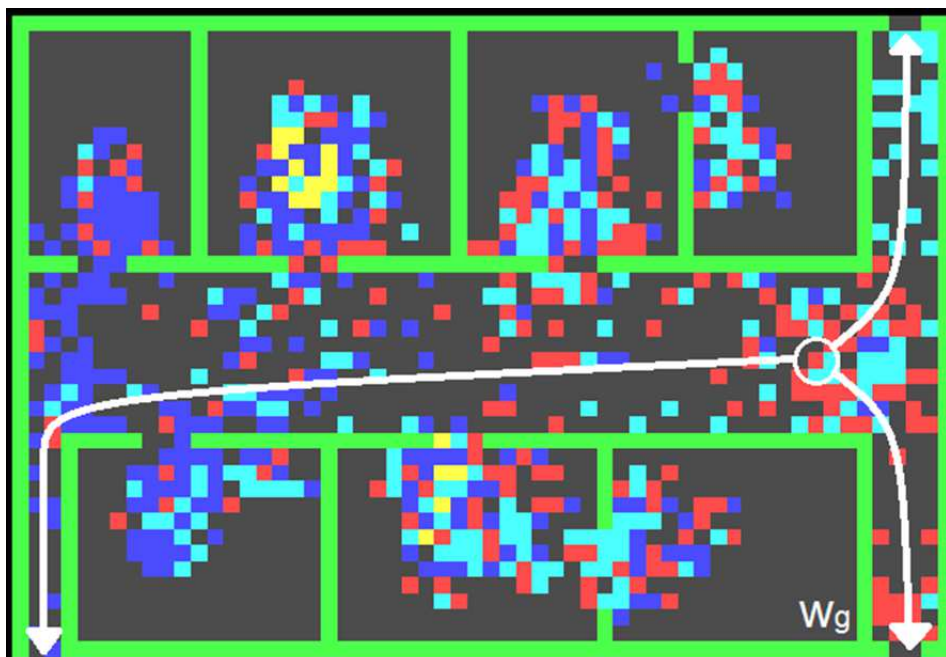


Figura 14: Representação dos CE usados no cálculo de μ . A célula em destaque que tem como referência a rota w_g encontrou em um congestionamento e terá que escolher uma nova rota (3 rotas possíveis). A probabilidade de seguir em cada direção será inversamente proporcional ao tamanho dos CE (distâncias), apresentados pelas linhas brancas. Neste caso a probabilidade de continuar na mesma rota será a de maior valor.

É válido que, em um congestionamento é praticamente impossível de seguir um caminho contrário a maioria dos pedestres. Esta teoria fornece uma outra regra para mudanças de rota: se a ocupação da vizinhança de uma célula na posição (i, j) for total e se todas as células (ou quase todas) na vizinhança n_z passos tiverem o mesmo CE, exceto a célula da posição (i, j) , então esta célula poderá mudar de CE. Esta regra está ilustrada na Figura 15.

Seja $S_{ij}(1)$ e $S_{ij}(2)$ os dois possíveis CE do autômato \mathcal{A} . Denominaremos como $S_{ij}^{(t)}$ o valor do CE no passo (t) . Suponha que, no passo (t) , a célula de referência tenha como orientação $S_{ij}(1)$ e que a sua vizinhança n_z passos tenha uma ocupação de células com

orientação $S_{ij}(2)$ maior ou igual a ζ . Logo, com probabilidade π , teremos $S_{ij}^{(t+1)} = S_{ij}(2)$ e com probabilidade $(1 - \pi)$ teremos $S_{ij}^{(t+1)} = S_{ij}(1)$.

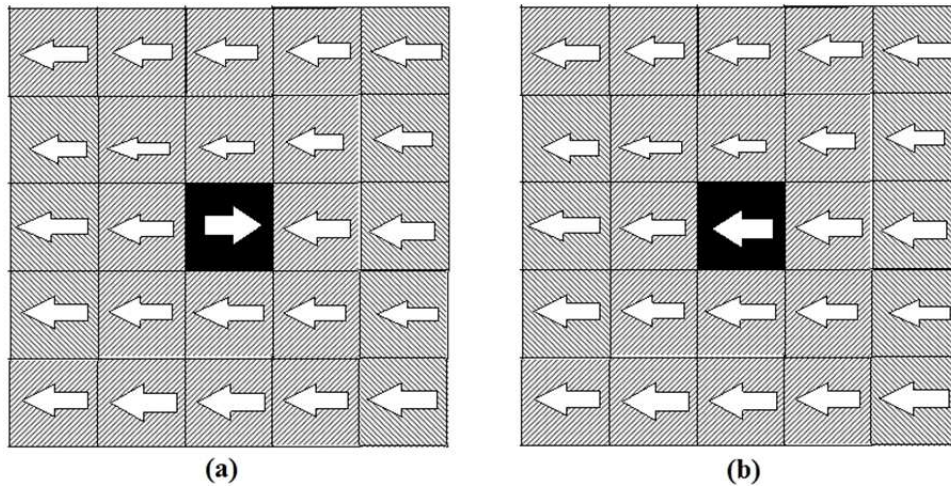


Figura 15: Representação da regra segunda de mudança de rota. As setas indicam o sentido preferencial determinado pelo CE. Em (a) apenas a célula central tem direção contrária a vizinhança. Em (b), com probabilidade π , o sentido preferencial é alterado devido a configuração das células ocupadas em uma vizinhança 2 passos.

3.3 Campo de Grupos

A partir da seção anterior, vimos que uma célula poderá alternar seu percurso de acordo com algumas circunstâncias importantes. A partir da adição desse novo elemento ao modelo de AC, surge uma questão importante. Nem todos pedestres agem alternando seu percurso por vontade própria mas sim por um comportamento coletivo. Exemplos típicos que ilustram esse comportamento são os grupos familiares, grupos escolares e grupos de amigos, dentre outros. A importância em considerar esses grupos é que eles podem ter influência direta nos parâmetros de segurança, já que variáveis como velocidade e tempo de evacuação podem sofrer alterações devido a influência direta

dos grupos. Elementos em grupos podem desencadear congestionamentos pelo fato de sempre caminharem em proximidade e/ou diminuírem a velocidade (podem até mesmo parar) quando algum elemento do grupo não está próximo.

Para criação dos grupos no modelo de AC, foi criado um novo campo de atração chamado *campo de grupo (CG)*. Este campo será uma área definida no espaço pelos próprios elementos do grupo, sendo que as coordenadas são determinadas pelo menores e maiores valores de linhas e colunas, dinamicamente a cada passo, conforme a Figura 16. Neste caso o modelo para cálculo das probabilidades de transição terá o acréscimo de uma variável indicadora, que anula as probabilidades de transição para posições fora do CG para todos os elementos do grupo. Dentro do CG o movimento é livre e será definido pelo CE. Todos os elementos do grupo obviamente terão o mesmo CE, que poderá ser alterado caso o grupo encontre algum congestionamento. A possível mudança de rota do grupo será definida pelo elemento mais adiantado, utilizando as mesmas regras adotadas anteriormente. Todos os outros elementos irão adotar a mesma postura.

Em alguns casos pode surgir algum obstáculo no meio do grupo que impede a progressão do mesmo até a saída. A Figura 17 ilustra um caso possível. Casos como este acabam criando uma parada definitiva do grupo pelo fato da área de atuação do CG não poder ser deslocada. Logo, uma solução encontrada foi a interrupção temporária do efeito do CG nos elementos que se encontram retidos pelo obstáculo. Com isso, estes elementos poderão então caminhar livremente sem o efeito do CG enquanto o grupo "espera" a aproximação dos mesmos. Assim que o elemento superar o obstáculo, o CG volta a fazer efeito sobre o mesmo e o grupo retorna o percurso até a saída. Existem casos em que os elementos não conseguem superar o obstáculo pelo fato de não exis-

tir caminho livre. Estes casos ocorrem quando outras células estão bloqueando todo o caminho, conforme apresentado na Figura 18. Eventos como estes foram observados com frequência nas simulações deste trabalho e podem ocorrer também com frequência em situações reais. O problema gerado é que cria-se um círculo vicioso em que, ao mesmo tempo que o grupo bloqueia o caminho das outras células enquanto "espera" a célula retida, estas outras células bloqueiam o caminho do elemento retido do grupo, então cria-se um colapso, aumentando demasiadamente o tempo de simulação ou fazendo com que a simulação nunca seja concluída. A solução encontrada neste caso é permitir que, em um certo momento, o elemento do grupo retido possa deixar de fazer parte do grupo definitivamente. Assim o grupo continuará o seu percurso permitindo a passagem das outras células e estas por sua vez, permitindo a passagem do elemento retido, que agora caminhará para a saída independente do grupo. Assim será fixada uma probabilidade Δ de que um elemento retido, deixe de fazer parte do grupo. A probabilidade Δ garante, pela Lei dos Grandes Números, que em algum momento haverá uma solução do colapso, com probabilidade 1.



Figura 16: Representação do campo de grupo (CG). A área clara representa o CG para um grupo composto de cinco elementos.

Seja a célula na posição (i^*, j^*) uma célula escolhida como sendo uma célula de grupo. A probabilidade de transição de uma célula da posição (i^*, j^*) para uma posição

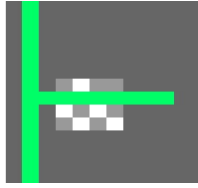


Figura 17: Representação de obstáculo que impossibilita movimentação do grupo. O elemento isolado terá a ação do CG interrompida.

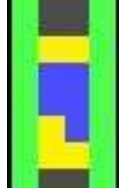


Figura 18: Representação de colapso provocado pelo CG. As células em cinza claro na parte inferior representa o grupo estagnado a espera das células na parte superior. As células em cinza escuro estão presas pela barreira formada pelo grupo.

(i, j) de sua vizinhança é dada por:

$$p_{ij} = N e^{K_d D_{ij}} e^{-K_a E_{ij}^{(m)}} \left(e^{K_s M_{ij}} \right)^{V_{ij}} \left(e^{K_s U_{ij}} \right)^{(1-V_{ij})} (1 - n_{ij}) \xi_{ij} I_{g1} \quad (3.7)$$

em que $I_{g1} = 0$ se a nova área do CG a ser gerado a partir da movimentação para a posição (i, j) é maior que η e $I_{g1} = 0$ caso contrário.

3.4 Condições Iniciais

Um fato importante e que pode impactar diretamente nas propriedades emergentes do sistema, são as condições iniciais de simulação. A condição mais importante e que será tratada aqui é a definição da rota de saída de cada célula no início da simulação. Por questões óbvias, é fácil chegar à conclusão que esta escolha deva ser aleatória, porém, com maiores chances da célula escolher a saída mais próxima, sendo que estas chances

devem aumentar de acordo com a proximidade da saída.

Para definição da rota ao qual a célula pertencerá no início da simulação, será utilizado como critério o CE, de forma que, a célula possuirá maior probabilidade de ter como saída aquela que fornece o menor CE.

Seja \mathcal{A} um automato celular estocástico para simulação de pedestres bidimensional definido com W campos estáticos. Seja $S_{ij}(w_p), p = 1, \dots, W$ um possível CE do autômato \mathcal{A} . Logo definimos

$$P(CE = S_{ij}(w_p)) = \left(\frac{\frac{1}{S_{ij}(w_p)}}{\sum_{l=1}^W \frac{1}{S_{ij}(l)}} \right) \quad (3.8)$$

sendo $P(CE = S_{ij}(w_p))$ a probabilidade do CE inicial para a célula na posição (i,j) ser igual a $S_{ij}(w_p)$.

3.5 Observações finais

Neste capítulo apresentamos o modelo proposto, que inclui as novas componentes de mudança de rotas e campos de grupos. Apresentamos inicialmente o novo formato para a probabilidade de transição que une os conceitos vistos em diferentes trabalhos. Primeiramente apresentamos as regras que tornam uma célula apta a escolher novas rotas, e em seguida, apresentamos as probabilidades de escolha de uma rota alternativa, que é baseada nas distâncias a ser percorrida. A seguir, vimos o conceito de campos de grupos, sua representação matemática no modelo e regras para resolução de conflitos provocados pelo campo. Por ultimo apresentamos as condições iniciais de escolha de rota que levam em consideração as distâncias a serem percorridas. Portanto, este capítulo fecha toda a parte teórica e começaremos a tratar a partir de agora, da parte técnica e prática, que consiste na criação do software de simulação e obtenção de dados para estudo do novo modelo.

4 SIMULAÇÕES E RESULTADOS

Este capítulo apresenta os resultados das simulações realizadas para avaliação do modelo de AC proposto. O principal objetivo das simulações é analisar o efeito de K_r e do número de grupos nas métricas de segurança, mais especificamente no tempo de evacuação total e na contagem de retenções, geradas a partir do novo modelo.

A validação de modelos de AC a partir de dados reais sempre foi uma questão de difícil solução devido ao fato de dados deste tipo serem de difícil acesso e a impossibilidade de se reproduzir eventos como este, ou seja, realizar simulações utilizando pessoas reais. Além disso, a reprodução destes eventos, principalmente em situação de emergência, pode não ser totalmente fiel aos acontecimentos reais devido ao fator psicológico envolvido.

No entanto, em Suma, Yanagisawa e Nishinari (2012), um experimento realizado com vinte e cinco participantes permitiu uma comparação entre simulações pelo modelo proposto (modelo de AC com efeito de antecipação) e os resultados deste experimento. O experimento consistia em dois grupos, que deveriam atravessar um mesmo corredor, ao mesmo tempo, a partir de lados contrários. A métrica utilizada foi o tempo médio gasto pelos participantes para atravessar o corredor. Num primeiro cenário pediu-se que os grupos andassem pelo corredor utilizando ao mesmo tempo um telefone celular

(baixo valor de CA). Em outro cenário, deveriam caminhar normalmente sem nenhuma instrução específica (médio valor de CA) e no último cenário deveriam caminhar com total atenção, sempre procurando se desviar de outros pedestres a frente (alto valor de CA).

Para a realização das simulações pelo modelo, três valores de K_r foram utilizados para representar os três cenários descritos. Obviamente, não foi possível fazer uma comparação quantitativa, de forma que apenas uma comparação qualitativa de tendência foi realizada. Os resultados apresentaram a mesma tendência, mostrando que o modelo proposto consegue representar bem os resultados do experimento.

Podemos dizer que o modelo proposto neste trabalho é o modelo proposto por Suma, Yanagisawa e Nishinari (2012) acrescido de três novas componentes: a mudança de trajetória, a mudança de velocidade e a presença de grupos. Assim é válido que estes são modelos encaixados, em que a estrutura básica permanece a mesma, e podemos então estender as validações feitas para este novo modelo.

A primeira parte do capítulo apresenta uma descrição da criação do ambiente a ser simulado. A segunda parte apresenta a primeira etapa de simulações, cujo objetivo foi de realizar uma avaliação qualitativa e/ou subjetiva.

A terceira parte deste capítulo apresenta a segunda etapa de simulações, que consiste numa comparação entre o modelo tradicional e o modelo proposto neste trabalho. Para isto utilizaremos um ambiente simples, que consiste em uma sala com 3 possíveis saídas (Figura 19), sendo que a distribuição inicial para todas as simulações seguirá o que foi apresentado na Subseção 3.4. Para as simulações, utilizaremos como principal métrica o tempo até a evacuação total (T), que consiste numa estimativa do tempo, baseada

no numero de passos discretos até a completa evacuação do ambiente a ser estudado, conforme Tregenza (1976).

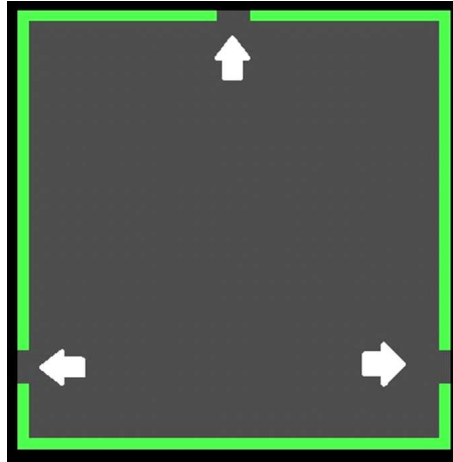


Figura 19: Ambiente utilizado na segunda etapa de simulações. As setas indicam as três saídas possíveis.

Na quarta parte deste capítulo são apresentados os resultados de simulações realizadas na terceira etapa de simulações, feitas em um ambiente complexo, com salas, corredores e 3 saídas (Figura 20). Neste ambiente, não foi possível realizar simulações para o modelo tradicional, conforme será discutido adiante. Para estas novas simulações, além do tempo T , outra informação a ser avaliada é a contagem de retenções (CR), que consiste em contar, para todas as células do autômato, o número de vezes que as mesmas ficam retidas (parada na mesma posição ou deslocamento contrario à saída) em cada simulação. O valor final de CR é a soma da contagem de retenções de todas as células ao fim da simulação. Esta medida pode ser importante para avaliar se a saída das células no ambiente está sendo contínua ou pausada, o que pode indicar gargalos ou obstáculos que interferem diretamente no tempo de evacuação. O objetivo desta terceira etapa é verificar o efeito dos novos parâmetros K_r e número de grupos em duas métricas

de segurança: o tempo T e a contagem de retenções CR.

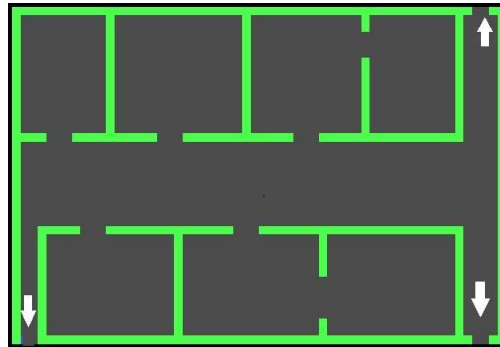


Figura 20: Ambiente utilizado primeira e terceira etapa de simulações. As setas indicam as três saídas possíveis.

Para os estudos de simulação, os parâmetros do modelo foram divididos em parâmetros fixos e variáveis. Os parâmetros fixos são aqueles que não sofrem mudanças ao longo das simulações, pelo fato de já terem sido estudados em outros trabalhos e por não representarem o foco principal deste trabalho. Os parâmetros variáveis apresentam diversos valores, dos quais será verificado o efeito dos mesmos em T . Na primeira etapa de simulações, os parâmetros variáveis são K_r e número de grupos. Na segunda etapa, além destes, temos também a lotação do ambiente como um parâmetro variável.

Os valores dos parâmetros utilizados para simulação encontram-se nas Tabelas 3 e 4. Lembramos que as simulações realizadas para $K_r = 0$ e sem formação de grupos equivale ao modelo tradicional, pela simples retirada do efeito das novas componentes. Os resultados foram obtidos a partir de 500 simulações para cada combinação de parâmetros. O programa foi escrito em linguagem C++ utilizando o compilador DEV C++ versão 4.9.9.2. Os gráficos foram gerados a partir do software Minitab® for Windows, versão 16.2.4.

Tabela 3: Parâmetros para segunda etapa de simulações.

Parâmetro	Valor
K_d	1
K_a	1
K_s	10
α	0,3
δ	0,1
Lotação	30%
K_r	0,00; 0,30;
π	0,8
ς	6
φ	2
n_z	1
Δ	0,999
η	16
<i>Grupos</i>	0; 5;

Tabela 4: Parâmetros para terceira etapa de simulações.

Parâmetro	Valor
K_d	1
K_a	1
K_s	10
α	0,3
δ	0,1
Lotação	10% ; 20%; 30%; 50%
K_r	0,01; 0,05; 0,15; 0,20; 0,25; 0,30; 0,5; 0,75; 1
π	0,8
ς	6
φ	2
n_z	1
Δ	0,999
η	16
<i>Grupos</i>	0; 2; 5;

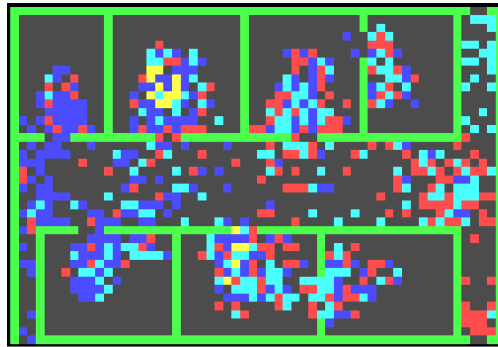


Figura 21: Foto do ambiente com lotação de 30% em início de simulação. As cores representam a intenção de saída. Células em cor azul escuro tendem a sair no canto inferior esquerdo, em azul claro tendem a sair no canto superior direito e em vermelho tendem a sair no canto inferior direito. Células em amarelo são células de grupo.

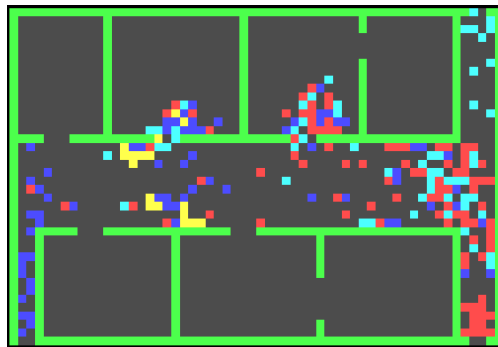


Figura 22: Foto do ambiente com lotação de 30% em estágio avançado de simulação.

4.1 Criação do Ambiente

Para a criação de um autômato a ser estudado pelo modelo, primeiramente deve-se criar o mapa com a ajuda de uma planilha eletrônica. Utilizamos neste trabalho o Software Livre LibreOffice versão 4.4.3.2. Em seguida, este mapa deve ser passado como parâmetro para o programa de simulação. Para construção do mapa, devemos fazê-lo através de uma codificação simples, conforme a Figura 23. As paredes do domínio devem possuir código 0 (zero). Os ambientes internos podem ser feitos com qualquer

código numérico, desde que cada ambiente tenha o seu próprio código. As fronteiras dos ambientes internos (portas, fim de corredor) possuem como código o numero 90 somado ao código do ambiente mais interno. O ambiente a ser gerado pela Figura 23, terá a saída definitiva onde se encontra o código 101. Como teremos mais de uma saída no mesmo autômato, deve-se criar uma matriz como esta para cada saída desejada, pois cada CE será criado com base na codificação feita. As cores são basicamente para melhor visualização. A cor cinza representa o interior do ambiente e a cor branca representa as paredes. O ambiente a ser criado pela Figura 23 está apresentado na Figura 20 e consiste no ambiente a ser estudado na primeira e terceira etapa de simulações.

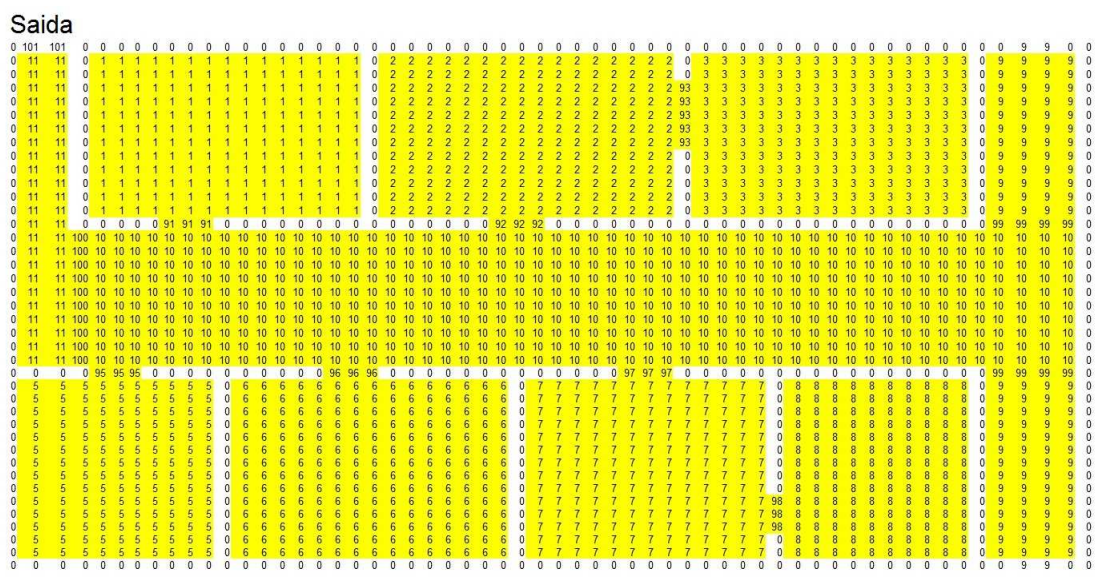


Figura 23: Planilha de criação do autômato.

4.2 Primeira Etapa de Simulações: Avaliação Subjetiva

Antes da realização das simulações para geração de dados, algumas simulações foram acompanhadas visualmente como uma forma subjetiva de validação. Além de uma

avaliação, esta parte do trabalho foi importante para definição de alguns parâmetros. As Figuras 22 e 21 apresentam fotos destas simulações em dois momentos distintos: início e estágio avançado, respectivamente.

Para uma melhor visualização das simulações, as cores de cada célula foram determinadas conforme sua saída preferencial ou conforme o seu CE, de forma que, ao mudar a direção, muda-se também a cor, exceto para as células de grupo que, independentemente da direção escolhida pelo grupo, permaneceu sempre a mesma. Esta diferença de cores foi uma forma encontrada de diferenciar as células por intenção de saída e agrupamento.

A seguir listamos os resultados desta avaliação:

- *Definição dos valores da matriz de preferências.* A matriz de preferências é vista como a distribuição a priori das probabilidades de transição. Modelos em que temos passeios aleatórios, a matriz de preferências deve conter valores de probabilidade iguais em toda matriz. Conforme é definindo um caminho preferencial, essas probabilidades se concentram em uma parte da matriz. Os valores utilizados para a matriz foram os mesmos usados em Pereira, Duczmal e Cruz (2013). A avaliação visual permitiu validar esses valores, pois os mesmos forneciam uma movimentação bastante natural das células, sem fornecer simulações em que há movimentos quase determinísticos ou quase aleatórios. Os valores utilizados para matrizes de preferências em vizinhança 1 passo e 2 passos são apresentados na Tabela 5
- *Procura por erros de programação.* A visualização da simulação é uma forma eficiente de verificar possíveis erros no algoritmo implementado. Qualquer ocor-

Tabela 5: Valores da matriz de preferências.

Matriz $N_M^1(c)$	Matriz $N_M^2(c)$
m1=0,4; a=0,01; b=0,05; c=0,2; d=0,02; e=0,06; f=0,01; g=0,05;	u1=0,26; a=0; b=0; c=0; d=0; e=0; f=0 g=0,03; h=0,03; i=0,05; j=0,07 k=0; l=0,03, m=0,1; n=0,25, o=0 p=0,03; q=0,03; r=0,05, s=0,07 t=0, u=0, v=0, x=0, z=0

rência anormal como células travadas, células em direção não muito bem definida, células que aparecem após a simulação ter começado ou ambiente mal desenhado deve ser verificado. Foram verificados alguns problemas como este nas primeiras simulações e em seguida foram completamente resolvidos, sendo constatado que se tratavam de erros de implementação e não de erros na metodologia.

- *Propriedades emergentes.* Algumas ocorrências emergentes também foram observadas durante esta etapa, como a formação de congestionamentos, colisões e movimentação. Foi visto que, conforme era esperado, os congestionamentos ocorrem com muita força nos corredores e nas salas internas. Além disso foi visto que as salas internas mais afastadas das saídas se esvaziam mais rápido, pelo fato do trânsito na sua porta ser menor. Quanto às colisões, observa-se que as células em lados opostos evitam contato direto, ficando evidente que o CA exerce influência positiva. Porém, em pontos mais densos, observa-se uma diminuição deste efeito, sendo quase nulo em pontos de maior concentração de células. Outra propriedade importante verificada nas simulações é o fato de não haver uma proporção na força contrária exercida pelas células que colidem. Nas simulações, é fácil ver que apenas duas células no meio de um corredor consegue conter ou obstruir a passagem de dezenas de células que estão em caminho contrário. Esta

desobstrução só ocorre quando estas células mudam de direção e resolvem seguir o mesmo caminho das demais, mas isto pode demorar um tempo razoável, já que isto não ocorre por pressão das outras células e sim pelo fato da mesma estar em um congestionamento. Este é um fato que surge a partir do novo modelo e que deve ser estudado em trabalhos futuros.

- *Avaliação das regras de mudança de rotas.* As simulações visuais também foram importantes para uma avaliação do modelo proposto. Durante as simulações, foi possível ver claramente as células que estavam em congestionamento mudarem suas direções por dois fatos marcantes: a mudança de cor e a saída do congestionamento. Além disso, também foi possível ver que estes fatos aumentavam conforme aumentava-se o valor de K_r . Fixando $K_r = 0$ foi possível ver a formação de retenções que nunca se dissolviam, principalmente nos corredores. Ficou claro que, quando não há probabilidade de mudança de direção (como nos modelos tradicionais) fica impossível obter simulações de evacuação com mais de uma rota de saída em ambientes que apresentam passagens estreitas como corredores, descobrindo a partir deste resultado, mais uma importante aplicação do modelo proposto.

Após o estudo visual, as simulações realizadas para obtenção dos dados estudados nas seções seguintes foram realizadas sem visualização, sendo esta uma forma de aumentar a velocidade de simulação.

4.3 Segunda Etapa de Simulações

A Tabela 6 fornece as estatísticas descritivas e intervalos de 95% de confiança para a média do tempo T relativo à primeira etapa de simulações. As estatísticas foram obtidas a partir das 500 simulações realizadas para cada configuração dos parâmetros variáveis, discriminados na primeira coluna. Os intervalos de confiança foram obtidos a partir teorema do limite central. De um modo geral, observar diferenças significantes entre o modelo proposto e o modelo tradicional para T pode representar uma correção efetiva nas estimativas dos parâmetros de segurança, proporcionado pela inclusão dos novos fatores. A principio, é lógico pensarmos que a falta do fator de decisão para mudança de direção pode estar superestimando a principal métrica de medida de segurança T. Ao mesmo tempo, a falta de grupos nas simulações subestimam essa mesma métrica.

Tabela 6: Estatísticas descritivas da segunda etapa de simulações.

Kr	Grupos	Média	Desv.Pad.	Mediana	Mínimo	Máximo	I.C.
0	0	2,5349	0,2764	2,4920	1,9727	3,0013	(2,5107;2,5591)
0,3	0	2,0378	0,1120	2,0373	1,7392	2,2460	(2,0280;2,0476)
0	5	2,9281	0,6759	2,7305	1,9777	5,3715	(2,8689;2,9874)
0,3	5	2,1467	0,1777	2,1218	1,7491	2,6485	(2,1311;2,1622)

A Tabela 6 apresentou resultados que seguem o que era esperado. Foi verificado uma diminuição nas médias e medianas de T quando aumenta-se K_r , de 0 para 0,3. Com relação ao numero de grupos, verifica-se um aumento acentuado das médias e medianas de T ao aumentar de 0 para 5 grupos. Com o aumento de K_r e de grupos simultaneamente, verifica-se uma diminuição nas medidas de posição de T. Os intervalos de confiança não se sobrepõem, indicando diferenças estatisticamente significantes para a média de T em todas as combinações estudadas. Com base nas médias, podemos dizer

que, o modelo tradicional apresentou uma diferença de aproximadamente 0,5 minutos em relação ao modelo com fator apenas de mudança de rotas. Esta diferença foi de 0,39 minutos em relação ao modelo com fator apenas de presença de grupos. Considerando o modelo completo, que possui os fatores de mudança de rotas e presença de grupos, essa diferença foi de 0,39 minutos. O desvio padrão obtido pelas simulações tiveram a mesma tendência apresentada pelas medidas de posição. Em relação ao modelo tradicional, há uma queda de dispersão na presença do fator de mudança de rotas K_r (diferença de aproximadamente $0,16 \text{ min}^2$) e um aumento de dispersão na presença de grupos (diferença de aproximadamente $0,4 \text{ min}^2$). Na presença destes fatores simultaneamente, observa-se uma queda de dispersão (diferença de aproximadamente $0,1 \text{ min}^2$). Estas mesmas tendências também foram válidas para as demais estatísticas. Essas diferenças podem ser interpretadas como uma estimativa do vício do modelo tradicional em não levar em consideração estes novos fatores. Lembramos que o objetivo destas simulações foram unicamente comparar o modelo tradicional ao modelo com a presença de grupos e efeito de mudança de rota, por isso utilizamos simulações feitas apenas para um único valor de K_s e para uma única quantidade de grupos. Os valores escolhidos de K_r e grupos seriam escolhidos subjetivamente até que as simulações apresentassem diferenças significativas entre o modelo tradicional e proposto. Os valores $K_r = 0,3$ e a quantidade de 5 grupos foram os primeiros a serem escolhidos e como houveram diferenças significativas em T, foram mantidos. Na segunda etapa de simulações será feito um estudo mais detalhado do efeito de K_r e do número de grupos.

4.4 Terceira Etapa de Simulações

A partir de agora, uma nova etapa de simulações foi realizada para se fazer uma análise mais detalhada dos novos parâmetros em um ambiente mais realístico e completo, além de verificar a influência dos mesmos em diferentes lotações. Espera-se que os efeitos destes novos parâmetros sejam mais evidentes em ambientes com alta lotação. O ambiente a ser simulado é apresentado na Figura 20, que consiste em um ambiente complexo, com 7 salas, 4 corredores e 3 saídas de emergência indicadas pelas setas. As medidas estimadas para o ambiente numa situação real são calculados conforme a escala apresentada por Tregenza (1976) e discutidas na Subseção 2.5.1. As Figuras 21 e 22 apresentam fotos do ambiente em dois momentos distintos de uma simulação, feita com lotação de 30% onde se vê claramente quatro cores distintas para as células. A Tabela 4 apresenta os novos valores dos parâmetros para esta etapa de simulação. note que a lotação agora passa a ser um parâmetro variável. Uma informação importante fornecida por essas novas simulações é o fato das simulações não poderem ser realizadas neste ambiente para $K_r = 0$, considerando a distribuição inicial dada na seção 3.4. Observando as simulações visualmente, foi verificado que o fato de existirem corredores estreitos e a impossibilidade de mudança de rota (pois $K_r = 0$) acabaram gerando congestionamentos que progrediram para o colapso do sistema (parada total). Isso mostrou mais uma importante contribuição para o modelo proposto: a correção de uma limitação dos modelos tradicionais em simulações de evacuação emergencial feitas em alguns ambientes complexos.

As Tabelas 7 e 8 apresentam as estatísticas descritivas das simulações realizadas e os intervalos de confiança de 95% para T e CR respectivamente. Em uma análise

conjunta com os gráficos das Figuras 24 e 25 fica evidente que, independente da lotação e do número de grupos, uma tendência geral é de queda considerável de T e CR nos primeiros valores de K_r , seguida de uma estabilização com o aumento de K_r . Os intervalos de confiança apresentam sobreposição a partir de $k_r = 0,05$ indicando que, somente os valores de K_r menores que 0,05 afetam consideravelmente T e CR. Com base na informação de que o T e CR são infinitos quando K_r é igual a zero (pois cria-se o colapso), temos que $T \rightarrow \infty$ e $CR \rightarrow \infty$ quando $K_r \rightarrow 0$. A conclusão direta que temos deste resultado é que quando supõe-se uma resistência muito forte para mudar de saídas, T e CR poderão ter valores muito altos, mas a partir de um nível de resistência mais fraco, os valores de T e CR passam a ter um valor constante (ou fazer parte de grupo de valores com diferenças muito pequenas). Este efeito parece ficar mais destacável com o aumento da lotação. Com relação à dispersão, uma análise do desvio padrão pelos gráficos mostra que há uma maior dispersão de T e CR para valores baixos de k_r , sendo que esta dispersão diminui com o aumento de K_r . Para o caso em que não temos a presença de grupos, observamos a estabilização da dispersão para $K_r \geq 0,05$. Quando temos o aumento do número de grupos, aumenta-se também a dispersão e esta torna-se inconstante com o aumento de K_r . Novamente o aumento da lotação torna estes efeitos mais nítidos. Estes efeitos, apesar de ser comum em T e CR, foram mais evidente em T. Isso mostra que a presença de grupos cria variações importantes e que podem influenciar significativamente o resultado de uma simulação. O aumento da dispersão pode representar um risco a mais para o ambiente e também podem ser tratado como uma medida de segurança. Este resultado mostra a importância de estudos preliminares sob a definição do número de grupos nos ambientes a serem simulados. Vale ressaltar também que a adição das novas componentes simultaneamente fornece resultados com

menor variabilidade, indicando um modelo com medidas mais precisas, sendo apenas isso um importante resultado.

Acredita-se que os novos parâmetros irão influenciar os resultados em qualquer ambiente, mas vale lembrar que os resultados apresentados aqui podem também sofrer alterações conforme o ambiente a ser simulado e que estes resultados são válidos apenas para o ambiente estudado aqui. Além disso este estudo se limitou a estudar apenas algumas combinações pontuais dos parâmetros K_r e número de grupos, devido ao fato de haver infinitas possibilidades. Assim, para outras combinações não estudadas aqui, podemos ter cenários bem diferentes. Além disso, algum parâmetro considerado fixo durante as simulações pode ter interação significativa com K_r e grupos, de forma que, alguma mudança nos mesmos pode criar efeitos consideráveis nos resultados apresentados.

4.5 Observações finais

Neste capítulo apresentamos o desenvolvimento de toda parte prática deste trabalho, que consistiu na criação do software, simulações e análise de resultados. Detalhamos cada parte deste processo, apresentando todas as etapas de simulações, resultados numéricos, gráficos e suas principais conclusões e contribuições. Este capítulo apresenta resultados importantes, enfatizando as possíveis correções proporcionadas pelo novo modelo nas medidas de segurança.

Tabela 7: Estatísticas descritivas para o tempo de evacuação (T).

Lotação	Grupos	Kr	Média	DP	Mediana	Min	Max	Int. de confiança (95%)	
10,00%	0	0,01	1,8370	0,4708	1,6795	1,2025	4,0547	1,7957	1,8782
		0,05	1,5112	0,1191	1,5006	1,1727	1,7789	1,5007	1,5216
		0,15	1,4736	0,1093	1,4659	1,2124	1,9180	1,4641	1,4832
		0,2	1,4867	0,1148	1,4808	1,2224	1,7889	1,4766	1,4968
		0,25	1,4712	0,1094	1,4609	1,2224	1,7193	1,4616	1,4808
		0,3	1,4812	0,1130	1,4708	1,2025	1,9031	1,4713	1,4911
		0,5	1,5078	0,1157	1,5056	1,2522	1,7193	1,4976	1,5179
		0,75	1,5255	0,1041	1,5255	1,2870	1,7640	1,5164	1,5347
		1	1,5255	0,1041	1,5255	1,2870	1,7640	1,5164	1,5347
		0,01	2,6792	2,1069	1,9031	1,2323	10,2263	2,4945	2,8639
	0,05	1,7854	0,5533	1,6100	1,2373	2,8075	1,7369	1,8339	
	0,15	1,5860	0,3487	1,5354	1,2075	1,7739	1,5554	1,6166	
	0,2	1,5890	0,3487	1,5354	1,2075	1,7739	1,5554	1,6166	
	0,25	1,5775	0,1777	1,5553	1,2671	1,8684	1,5619	1,5931	
	0,3	1,5775	0,1777	1,5553	1,2671	1,8684	1,5619	1,5931	
	0,5	1,5753	0,1930	1,5429	1,2075	2,3156	1,5584	1,5922	
	0,75	1,5962	0,1415	1,5851	1,2770	2,0224	1,5838	1,6086	
	1	1,6030	0,1400	1,5901	1,2919	1,8684	1,5908	1,6153	
	0,01	3,9846	3,5515	2,5243	1,3218	24,9942	3,6733	4,2959	
	0,05	1,9863	1,2275	1,8977	1,2069	24,9032	2,3039	2,3039	
	0,15	1,8112	0,5197	1,6944	1,2572	2,6982	1,7656	1,8568	
	0,2	1,7833	0,8235	1,6770	1,2621	2,5789	1,7112	1,8555	
	0,25	1,7353	0,3029	1,6671	1,3267	3,2249	1,7087	1,7618	
	0,3	1,7394	0,7515	1,6497	1,2969	2,9615	1,6735	1,8053	
0,5	1,6953	0,3124	1,6597	1,3416	1,9528	1,6679	1,7227		
0,75	1,6998	0,1900	1,6646	1,3466	1,9528	1,6831	1,7164		
1	1,7181	0,1814	1,6845	1,3565	2,5441	1,7021	1,6750		
0,01	4,5781	1,3440	4,4542	2,4646	7,6622	4,4603	4,6959		
0,05	2,8432	0,1749	2,8224	2,3653	3,2150	2,8279	2,8585		
0,15	2,6705	0,1338	2,6634	2,3454	2,9516	2,6588	2,6823		
0,2	2,6451	0,1284	2,6286	2,2559	3,0311	2,6338	2,6564		
0,25	2,6413	0,1239	2,6336	2,3454	2,9615	2,6304	2,6521		
0,3	2,6413	0,1239	2,6336	2,3454	2,9615	2,6304	2,6521		
0,5	2,6486	0,1250	2,6355	2,3205	3,0907	2,6304	2,6596		
0,75	2,6790	0,1198	2,6783	2,3603	2,9963	2,6685	2,6965		
1	2,7179	0,1140	2,7131	2,3901	2,9615	2,7079	2,7279		
0,01	5,9736	2,9004	5,2771	2,6584	16,1195	5,7194	6,2279		
0,05	3,3715	0,9737	3,0386	2,5591	13,2673	3,2862	3,4569		
0,15	2,9225	0,4611	2,8075	2,3752	3,5330	2,8821	2,9629		
0,2	2,9790	1,5710	2,7628	2,3007	4,1740	2,8413	3,1167		
0,25	2,9790	1,5710	2,7628	2,3007	4,1740	2,8413	3,1167		
0,3	2,8340	0,6346	2,8249	2,3389	2,9380	2,8378	2,8895		
0,5	2,7881	0,3566	2,7355	2,4000	3,3789	2,7568	2,8193		
0,75	2,8879	1,1437	2,7628	2,3802	3,0758	2,7877	2,9881		
1	2,8719	0,7297	2,7777	2,4895	3,3988	2,8079	2,9358		
0,01	7,2177	3,6839	6,9070	2,6584	16,4525	7,3988	8,0446		
0,05	4,1505	1,7604	3,7516	2,5690	6,6486	3,9962	4,3048		
0,15	3,3844	1,4600	3,0386	2,4646	24,9942	3,2564	3,5124		
0,2	3,3110	1,8203	2,9417	2,4646	12,1055	3,1514	3,5011		
0,25	3,3305	1,9467	2,9615	2,4348	8,2337	3,1598	3,5011		
0,3	3,2428	1,6465	2,9292	2,4398	4,5169	3,0985	3,3871		
0,5	3,1982	1,5253	2,9044	2,4696	4,0647	3,0645	3,3319		
0,75	3,1982	1,5253	2,9044	2,4696	4,0647	3,0645	3,3319		
1	3,1895	1,3781	2,9367	2,5441	5,6846	3,0687	3,3103		
0,01	7,5342	1,5913	7,5554	4,0895	10,3853	7,3947	7,6737		
0,05	4,1482	0,2170	4,1491	3,5181	4,1292	4,1292	4,1292		
0,15	3,8665	0,1464	3,8510	3,4783	4,1541	3,8537	3,8793		
0,2	3,8299	0,1527	3,8237	3,4584	4,0398	3,8165	3,8433		
0,25	3,8207	0,1488	3,8162	3,4286	4,1591	3,8076	3,8337		
0,3	3,7994	0,1340	3,7963	3,4734	4,1442	3,7876	3,8111		
0,5	3,8230	0,1312	3,8162	3,4485	4,2783	3,8115	3,8345		
0,75	3,8526	0,1334	3,8411	3,5081	4,0299	3,8409	3,8643		
1	3,9106	0,1250	3,9057	3,5259	4,1790	3,8986	3,9215		
0,01	8,4569	3,1525	8,0598	3,7367	17,4363	8,1805	8,7332		
0,05	4,9441	1,7333	4,4448	3,7218	24,9942	4,7922	5,0960		
0,15	4,3179	1,3135	4,0150	3,5429	6,6038	4,2028	4,4331		
0,2	4,3182	1,6314	3,9901	3,5032	6,4399	4,1752	4,4612		
0,25	4,3182	1,6314	3,9901	3,5032	6,4399	4,1752	4,4612		
0,3	4,2152	1,4655	3,9206	3,5479	10,7580	4,0867	4,3437		
0,5	4,2151	1,7937	3,9483	3,4833	5,2025	4,1292	4,3723		
0,75	4,1790	1,3247	3,9802	3,6324	4,8498	4,0629	4,2952		
1	4,2498	1,4301	3,9976	3,6622	9,4610	4,1245	4,3752		
0,01	10,5734	4,1708	9,4759	4,1044	24,3035	10,2078	10,9390		
0,05	5,8489	2,4543	5,1752	3,7019	13,3468	5,6337	6,0640		
0,15	5,0577	2,3949	4,4547	3,6175	19,6227	4,8477	5,2676		
0,2	4,8554	2,1451	4,2485	3,5578	8,0548	4,6674	5,0434		
0,25	4,6731	1,7934	4,2411	3,6274	10,5741	4,5159	4,8303		
0,3	4,7732	2,4109	4,1790	3,5678	12,6412	4,5619	4,9845		
0,5	4,7322	2,3951	4,1740	3,5727	13,0586	4,5223	4,9422		
0,75	4,6916	2,1814	4,1740	3,5827	19,7469	4,5004	4,8828		
1	4,8164	2,2461	4,1963	3,7218	15,6077	4,6195	5,0133		
0,01	12,6098	2,0114	12,5468	7,0113	18,0376	12,4335	12,7861		
0,05	6,8186	0,2478	6,8125	6,1467	7,5629	6,7969	6,8403		
0,15	6,2882	0,1695	6,2868	5,7989	6,5889	6,2734	6,3031		
0,2	6,2323	0,1661	6,2262	5,7541	6,8175	6,2177	6,2469		
0,25	6,2019	0,1641	6,1914	5,7740	6,4548	6,1875	6,2163		
0,3	6,2069	0,1683	6,2063	5,7144	6,6734	6,1922	6,2217		
0,5	6,2654	0,1997	6,2461	5,8038	6,8771	6,2479	6,2829		
0,75	6,3700	0,1842	6,3653	5,8982	6,6933	6,3539	6,3862		
1	6,4789	0,1835	6,4722	6,0722	6,9566	6,4628	6,4950		
0,01	13,9104	3,3812	13,3940	7,1902	24,9942	13,6141	14,2068		
0,05	7,7292	1,8209	7,0709	6,1914	11,7468	7,5696	7,8888		
0,15	7,0733	2,1817	6,4896	5,9032	13,1332	6,8821	7,2645		
0,2	6,8884	2,1905	6,3728	5,8187	9,7443	6,6964	7,0804		
0,25	6,8884	2,1905	6,3728	5,8187	9,7443	6,6964	7,0804		
0,3	6,7989	1,7695	6,3653	5,9231	13,0040	6,6438	6,9540		
0,5	6,7311	1,4869	6,3927	5,9380	14,8077	6,6008	6,8615		
0,75	6,8126	1,3659	6,5045	5,9628	24,7458	6,5929	6,9324		
1	6,8978	1,3825	6,6113	6,0871	24,9942	6,7766	7,0189		
0,01	15,3206	4,0951	14,5941	7,7815	24,9942	14,9617	15,6796		
0,05	9,1049	3,1951	8,1368	6,3455	24,9942	8,8248	9,3849		
0,15	7,7420	2,2862	7,0684	5,9380	11,0611	7,5416	7,9423		
0,2	7,6992	2,6153	6,7753	6,0225	13,9928	7,4699	7,9284		
0,25	7,7611	3,1467	6,8250	5,8784	14,2015	7,4853	8,0369		
0,3	7,5991	2,6945	6,7430	5,9032	9,1778	7,3629	7,8353		
0,5	7,9015	3,2934	6,7976	5,9231	12,8599	7,6128	8,1902		
0,75	7,4019	2,2026	6,7952	5,8635	24,9942	7,2088	7,5950		
1	7,6572	2,7248	6,8598	6,1616	21,0737	7,4183	7,8960		

Tabela 8: Estatísticas descritivas para a contagem de retenções (CR).

Lotação	Grupos	Kr	Média	DP	Mediana	Min	Max	Int. de confiança (95%)	
10.00%	0	0.01	18145,97	3160,77	17261,50	13578,00	36601,00	17868,92	18423,03
		0.05	16018,43	1024,88	16003,50	13511,00	17695,00	15928,59	16108,26
		0.15	15550,85	823,76	15530,00	13459,00	17096,00	15478,65	15623,06
		0.2	15451,82	851,49	15422,50	13144,00	18686,00	15377,18	15526,45
		0.25	15296,79	791,39	15273,00	13264,00	17852,00	15227,42	15366,15
		0.3	15375,26	809,05	15342,00	13265,00	17680,00	15304,35	15446,18
		0.5	15348,72	708,26	15367,00	13542,00	16668,00	15286,53	15110,80
		0.75	15459,87	732,11	15386,50	13767,00	16779,00	15395,70	15524,04
		1	15459,87	732,11	15385,50	13767,00	16779,00	15395,70	15524,04
		0.01	21690,59	8361,11	18442,50	13725,00	45839,00	20957,71	22423,48
	0.05	17571,74	2646,12	16851,00	13780,00	23136,00	17339,80	17803,68	
	0.15	16495,43	1596,42	16188,50	13111,00	18730,00	16355,50	16635,36	
	0.2	16295,04	1875,09	15996,50	13568,00	17519,00	16130,69	16459,40	
	0.25	16193,47	1365,75	16228,00	13892,00	25689,00	16073,75	16313,18	
	0.3	16193,47	1385,75	15970,00	13220,00	19207,00	16073,75	16313,18	
0.5	16094,29	1266,53	15881,50	13355,00	18923,00	15983,28	16205,31		
0.75	16125,18	1065,95	16028,50	13599,00	17543,00	16031,75	16218,62		
1	16145,02	979,09	16046,00	13861,00	18641,00	16059,20	16230,84		
0.01	0.01	27904,58	14066,21	22561,50	14307,00	65099,00	26671,62	29137,54	
	0.05	20060,17	4581,99	18958,50	14649,00	75047,00	19658,54	20461,80	
	0.15	17935,24	2545,03	17228,00	13892,00	25689,00	17712,16	18158,32	
	0.2	17803,41	2681,30	17220,00	14019,00	22613,00	17568,39	18038,44	
	0.25	17529,20	2038,10	17056,50	13696,00	23501,00	17350,55	17707,84	
	0.3	17364,06	2477,13	16886,00	13947,00	21517,00	17146,93	17581,19	
	0.5	17092,39	1653,40	16820,00	14247,00	19209,00	16947,46	17237,32	
	0.75	17158,40	1584,41	16920,50	14447,00	20581,00	17019,52	17297,28	
	1	17163,52	1348,85	17033,00	14201,00	24092,00	17045,29	17281,76	
	20.00%	0	0.01	84560,60	19303,87	80606,00	53402,00	134316,00	82868,54
0.05			61799,79	3386,64	61642,00	51187,00	71142,00	61502,94	62098,64
0.15			57686,52	2474,21	57532,00	51249,00	61460,00	57469,64	57903,39
0.2			57023,00	2310,99	57018,50	51587,00	61570,00	56820,43	57225,57
0.25			56564,33	2178,41	56614,50	49724,00	62564,00	56373,38	56755,27
0.3			56564,33	2178,41	56614,50	49724,00	62564,00	56373,38	56755,27
0.5			56375,67	2086,74	56215,00	51068,00	63576,00	56192,76	56558,58
0.75			59706,86	2052,29	59594,00	53984,00	62200,00	56526,97	56896,75
1			57280,81	1948,71	57188,00	52089,00	61388,00	57110,00	57451,62
0.01			100213,70	33539,74	91770,00	55510,00	204476,00	97273,81	103153,58
0.05		67010,73	9294,70	64514,50	54012,00	138555,00	66196,02	67825,45	
0.15		60991,33	5865,84	59413,00	52858,00	77669,00	60477,16	61505,49	
0.2		60962,19	11346,69	59003,00	52388,00	83591,00	59967,61	61956,77	
0.25		60962,19	11346,69	59003,00	52388,00	83591,00	59967,61	61956,77	
0.3		59043,58	5376,84	58652,00	53402,00	81112,00	58572,28	59514,88	
0.5	58449,16	3612,13	57730,00	52367,00	64069,00	58132,55	58765,78		
0.75	59099,03	7836,61	58180,50	51554,00	68256,00	58412,12	59785,94		
1	59233,19	5160,82	58437,00	52315,00	63603,00	58780,82	59685,55		
5	0.01	120721,80	42721,35	111197,50	56959,00	271703,00	116977,11	124466,49	
	0.05	75447,80	14455,76	72457,50	54680,00	120650,00	74180,70	76714,91	
	0.15	66919,18	12394,52	63977,50	53634,00	213647,00	65832,75	68005,60	
	0.2	65057,23	13665,87	62068,50	53848,00	131405,00	62655,10	64255,10	
	0.25	65422,64	15192,69	62068,50	51691,00	100003,00	64090,94	66754,34	
	0.3	64139,67	12589,40	61241,00	52265,00	84402,00	63036,16	65243,18	
	0.5	63178,98	11850,62	60736,50	53323,00	83223,00	62140,22	64217,73	
	0.75	63178,98	11850,62	60736,50	53323,00	83223,00	62140,22	64217,73	
	1	63010,96	9677,53	61206,00	52896,00	80794,00	62162,68	63859,23	
	30.00%	0	0.01	207325,06	37174,60	206520,50	137141,00	281087,50	204066,56
0.05			13635,31	6924,87	136823,31	117293,00	136748,00	136748,00	137962,30
0.15			127438,37	4277,42	127208,50	115686,00	135262,00	127063,44	127813,30
0.2			126048,11	3964,84	126096,50	115564,00	133943,00	125700,57	126395,64
0.25			125253,57	4119,73	125294,00	112528,00	135991,00	124892,46	125614,68
0.3			124719,73	3972,96	124685,00	114340,00	135651,00	124371,49	125067,98
0.5			124736,87	3836,62	124720,50	112608,00	134928,00	124400,68	125073,17
0.75			125347,97	3838,95	125334,00	113555,00	129649,00	125011,47	125684,47
1			126809,12	3984,42	126592,00	117827,00	131787,00	126594,93	127123,31
0.01			230563,79	69589,66	218193,50	127572,00	433068,00	224463,99	236663,60
0.05		148300,64	21220,62	143236,50	120532,00	283611,00	146440,57	150160,71	
0.15		135772,43	15585,54	131446,50	117654,00	153160,00	134406,29	137138,56	
0.2		134521,61	19096,34	129923,00	115087,00	183446,00	132847,74	136195,48	
0.25		134521,61	19096,34	129923,00	115087,00	183446,00	132847,74	136195,48	
0.3		131380,44	14451,56	127598,00	115867,00	198449,00	130113,71	132647,18	
0.5	131071,52	14874,85	127365,00	116725,00	195065,00	130375,55	132766,85		
0.75	130929,50	15994,45	128599,00	115410,00	145537,00	129527,53	132331,48		
1	131994,97	14482,28	129511,50	118019,00	185210,00	130725,54	133264,39		
50.00%	0	0.01	273507,22	88234,04	252079,50	131748,00	440952,00	265773,16	281241,27
		0.05	164120,79	30484,12	155136,00	122926,00	217196,00	161448,74	166792,84
		0.15	147245,26	25219,26	140169,50	118745,00	236278,00	145034,70	149455,83
		0.2	143318,38	21624,94	136982,50	115792,00	175727,00	141422,87	145213,89
		0.25	141320,20	21467,89	135524,00	117025,00	183444,00	139438,46	143201,95
		0.3	141358,33	25153,35	133859,00	115423,00	192529,00	139153,54	143563,12
		0.5	140186,14	25068,08	133115,50	117278,00	198300,00	137988,83	142383,45
		0.75	138061,78	19980,96	133583,00	117853,00	249497,00	136310,37	139813,19
		1	140049,45	21407,24	133969,00	120180,00	243678,00	138173,02	141925,88
		0.01	599701,50	87640,12	590272,00	363479,00	891074,00	592019,51	607383,50
	0.05	380002,25	14105,26	380117,00	335025,00	418011,00	378765,87	381238,63	
	0.15	351800,58	8984,28	351377,00	326493,00	384505,00	351022,07	352597,08	
	0.2	348319,94	8860,47	348456,50	320626,00	383488,00	347543,29	349096,60	
	0.25	345984,20	8545,77	345491,00	319732,00	373312,00	345235,13	346733,27	
	0.3	345571,91	8706,85	345099,00	322980,00	385105,00	344808,72	346335,10	
0.5	345493,83	9188,88	344517,50	320078,00	450960,00	344688,39	346299,27		
0.75	347034,04	7918,79	347444,00	327179,00	370309,00	346339,93	347728,16		
1	351968,83	7847,34	351907,00	330284,00	375576,00	351280,98	352656,68		
5	0.01	659072,56	139741,55	634141,50	399642,00	1052659,00	646823,67	671321,45	
	0.05	401972,12	45736,44	390401,50	339390,00	573517,00	397963,14	405861,10	
	0.15	371454,53	36852,49	359679,50	327943,00	463006,00	368224,27	374684,80	
	0.2	364181,15	38872,74	354281,00	320570,00	475614,00	360773,81	367588,50	
	0.25	364181,15	38872,74	354281,00	320570,00	475614,00	360773,81	367588,50	
	0.3	361314,93	35160,93	351995,50	322645,00	433720,00	358232,94	364396,92	
	0.5	358587,76	30204,88	350941,00	325761,00	439484,00	355940,19	361235,34	
	0.75	359353,00	22602,88	354265,00	322507,00	467608,00	357371,77	361334,23	
	1	362131,48	24700,84	357880,50	328804,00	715041,00	359966,35	364296,60	
	0.01	725502,64	181960,87	692683,00	408336,00	993967,00	709553,07	741452,21	
	0.05	439200,88	73659,80	413344,00	353132,00	881480,00	432744,31	445657,44	
	0.15	394258,62	51182,92	376205,50	332434,00	533203,00	389772,24	398745,00	
	0.2	385568,67	43930,90	368543,00	331757,00	463299,00	387717,96	389419,	

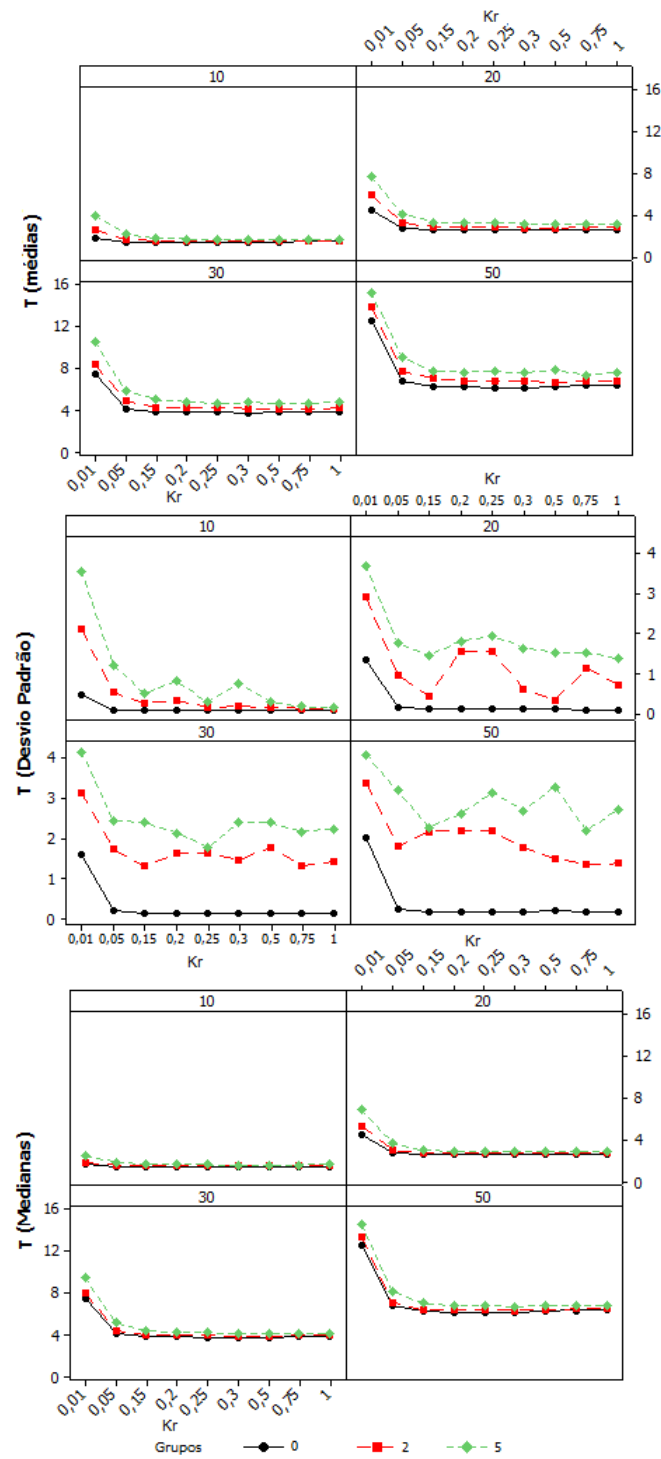


Figura 24: Gráficos de média, desvio padrão e mediana para o tempo de evacuação (T).

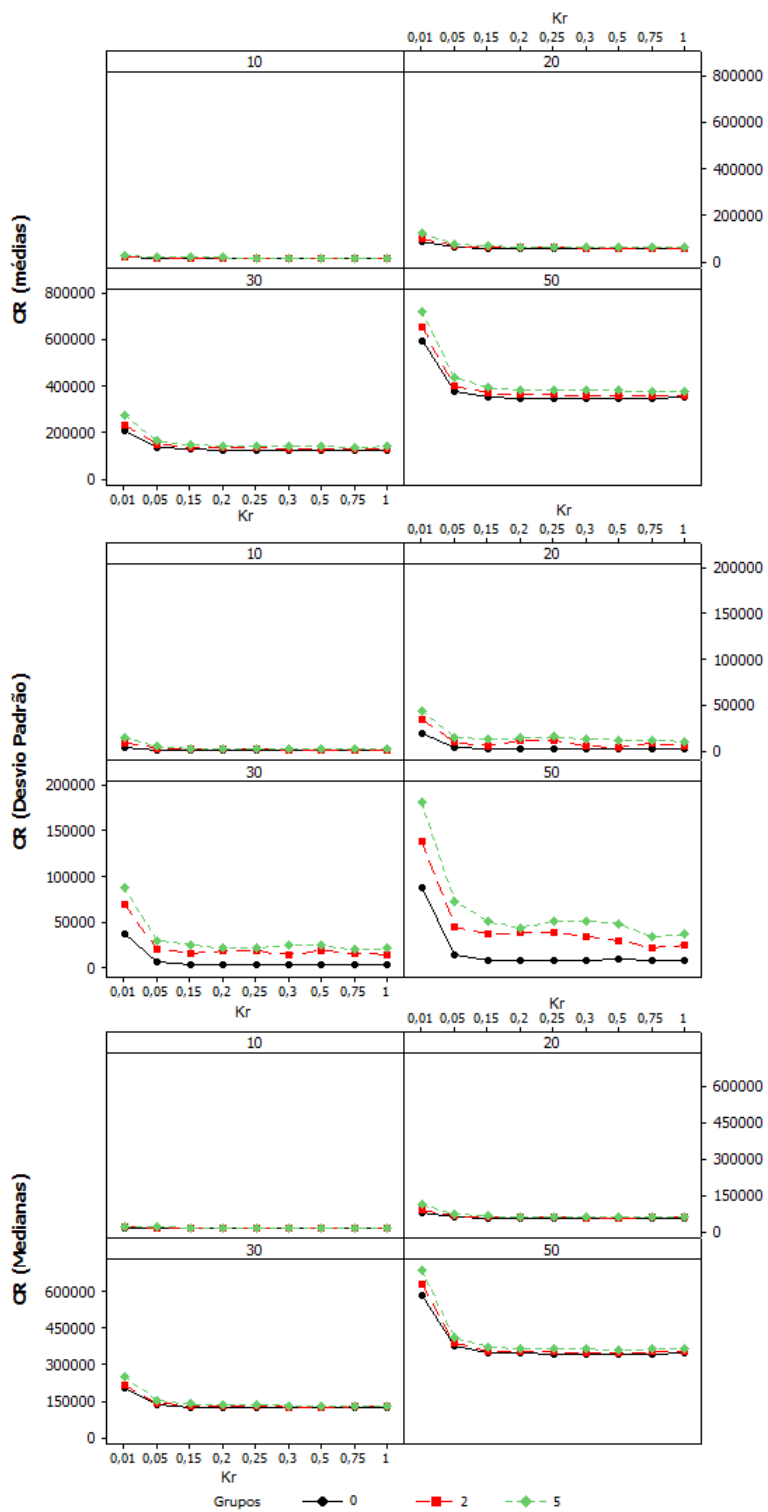


Figura 25: Gráficos de média, desvio Padrão e mediana para a contagem de retenções (CR).

5 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho teve como objetivo apresentar um novo modelo para simulação de ambientes em estado de evacuação via autômatos celulares não determinísticos.

Primeiramente apresentamos uma introdução e a revisão bibliográfica acerca do tema. Os principais objetivos desta parte foram apresentar as razões e/ou justificativas para o desenvolvimento deste trabalho. Vimos que os modelos existentes até o momento não abordam problemas importantes como a escolha e a mudança de direção e que estes podem ser importantes para realização de estudos em ambientes complexos, com mais de uma rota de saída. A revisão bibliográfica foi feita de forma a situar a nossa teoria na linha de tempo, além de apresentar as principais contribuições, até a chegada da atual fronteira do conhecimento.

Em seguida os fundamentos teóricos foram desenvolvidos para mostrar os principais tipos de modelagem de dinâmica de pedestres. Vimos uma breve introdução de cada uma delas e que nenhuma das teorias pré-existentes apresentam estudos que envolvem ambientes com mais de uma rota de saída. Ao apresentar a teoria de AC, houve um detalhamento maior para entendimento futuro da teoria a ser tratada.

Na próxima seção foi apresentado os AC para dinâmica de pedestres e evacuação

Emergencial. Vimos um detalhamento dos principais componentes do modelos desenvolvidos até aqui. Vimos também as diversos formatos da probabilidade de transição no tempo, com o objetivo de apresentar a sua evolução e de termos um melhor entendimento das expressões mais atuais.

No capítulo seguinte, apresentamos o modelo proposto com a sua probabilidade de transição, as regras para mudança de direção e o campo de grupos. Toda a estrutura do novo modelo foi detalhada.

A seguir algumas simulações foram realizadas com os objetivos de verificar o efeito das novas componentes nos resultados fornecidos pelo modelo proposto. As primeiras simulações comparam o modelo sem os novos componentes e o modelo proposto, incluindo as novas componentes. além disso uma segunda etapa de simulações faz um estudo comparando diversas combinações dos parâmetros das novas componentes. Os resultados observados forneceram as seguintes conclusões:

- Há diferenças significantes no tempo de evacuação e retenções quando o modelo apresenta as novas componentes, e a direção dessas diferenças se apresenta conforme era esperado. Este resultado é um forte indício de que pode haver vícios de estimativa nos modelos tradicionais, quando não levamos em conta fatores importantes como estes que foram apresentados e adicionados no modelo. Dessa forma o modelo proposto visa fornecer estimativas mais confiáveis, permitindo que os estudos realizados possam elevar o nível de segurança final dos ambientes reais. Estas conclusões são mais evidentes para ambientes com maior demanda de ocupação.
- Além disso vimos que em alguns ambientes complexos com mais de uma saída

pode não ser possível utilizar os modelos tradicionais. Um exemplo disto é o próprio ambiente analisando na segunda etapa de simulações. A possibilidade de mudança de direção apresentada aqui resolve este problema, aumentando em grande escala o conjunto de ambientes que podem ser estudados via AC.

- A segunda etapa de simulações mostrou que o modelo é sensível a mudanças em K_r que altera as probabilidades de mudança de rota. Vimos que o tempo de evacuação e retenções tendem a ser maiores quando temos baixa probabilidade de mudança de rotas ($K_r < 0,05$) e que quando aumentamos a probabilidade de mudanças de rota ($K_r > 0,05$), estas medidas de segurança tendem a permanecer constantes. Isso pode indicar que em ambientes onde espera-se poucas indecisões, deve haver uma demora maior para a evacuação. Isso fica mais evidente em ambientes com maior ocupação.

Deve-se lembrar que estas conclusões podem ser restritas ao ambiente estudado, de forma que podem variar conforme o ambiente a ser estudado. Dessa forma, antes de se utilizar o modelo para simulações de ambientes reais, o pesquisador deve realizar estudos preliminares de forma a obter uma parametrização mais eficiente possível. Exemplos de conhecimentos que podem ser feitos para este fim são:

- Tipo de público que usará o ambiente quanto a idade. Pessoas mais idosas tendem a ter movimentos mais lentos enquanto que Jovens tendem a ter movimentos mais rápidos. Os parâmetros ligados a esta característica são K_a e matriz de preferências.
- Conhecimento do público sobre o ambiente. Sabe-se que em um ambiente empre-

sarial, que é bem conhecido por todos que o utilizam, a evacuação tende a ser mais rápida que em uma galeria de compras, por exemplo. Nestas galerias, a tendência é de haver um rastro mais forte, já que muitos utilizadores irão seguir outros, pelo fato de não conhecer o ambiente. Os parâmetros envolvidos neste caso são o K_e , K_d e K_r .

- Finalidade do ambiente para o público. Em uma escola, por exemplo, haverá uma formação maior de grupos que em um Supermercado. Neste caso o parâmetro envolvido é o número de grupos.
- Lotação do ambiente. Escolas, casa de Shows, ginásios apresentam alta lotação se comparados a uma galeria ou um prédio residencial. O parâmetro ligado a esta característica é a lotação.

Para trabalhos futuros pretende-se realizar o acréscimo de novas componentes que levam em consideração a pressão ou força imposta por aglomerados de pedestre em sentidos opostos ou por aglomerados contra grupos estagnados. Vimos nas simulações, que a estagnação de apenas um par de pedestres pode bloquear grandes aglomerações por um tempo razoável. Talvez seja adequado considerar esta força nas probabilidades de mudança de rotas. Pretende-se também criar um modelo que diversifique as populações, como por exemplo, diferenciar as populações quanto ao sexo e à idade. Isso será feito fornecendo valores de parâmetros diferenciados para cada população, garantindo a heterogeneidade que pode influenciar nas simulações.

REFERÊNCIAS

- ALONSO, E. J. A. *Autômatos Celulares Generalizados como Modelos de Influência para Agrupamentos de Dados e Interações Sociais*. Tese (Tese de Doutorado) — COPPE - Universidade Federal do Rio de Janeiro, 2008.
- BURSTEDDE, C.; KLAUCK, K.; SCHADSCHNEIDER, A.; ZITTARTZ, J. Simulation of pedestrian dynamics using a two-dimensional cellular automaton. *Physica A: Statistical Mechanics and its Applications*, v. 295, n. 3-4, p. 507–525, 2001.
- CAMAZINE, S.; DENEUBOURG, J. L.; FRANKS, N. R.; SNEYD, J.; THERAULAZ, J.; BONABEAU, G. *Self-organization in Biological Systems*. United States: Princeton University Press, 2003.
- CRUZ, F. R. B.; MACGREGOR SMITH, J.; MEDEIROS, R. O. An M/G/C/C state dependent network simulation model. *Computers & Operations Research*, v. 32, n. 4, p. 919–941, 2005.
- CRUZ, F. R. B.; OLIVEIRA, P. C.; DUCZMAL, L. State-dependent stochastic mobility model in mobile communication networks. *Simulation Modelling Practice & Theory*, v. 18, n. 3, p. 348–365, 2010.
- FERBER, J. Foundations of distributed artificial intelligence. In: _____. New York, NY: John Wiley & Sons, 1996. cap. Reactive Distributed Artificial Intelligence: Principles and Applications, p. 287–314.
- FERBER, J. *An Introduction to Distributed Artificial Intelligence*. New York, NY: Addison-Wesley Longman, 1999.
- GARDNER, M. Mathematical games: The fantastic combinations of John Conway's new solitaire game 'life'. *Scientific American*, v. 223, p. 120–123, 1970.
- HAMAGAMI, T.; HIRATA, H. Method of crowd simulation by using multiagent on cellular automata. In: IEEE. *IEEE/WIC International Conference on Intelligent Agent Technology - IAT 2003*. [S.l.], 2003. p. 46–52.
- HELBING, D.; MOLNÁR, P. Social force model for pedestrian dynamics. *Physical Review E*, American Physical Society, v. 51, p. 4282–4286, 1995.

- ILACHINSKI, A. *Cellular Automata: A Discrete Universe*. Singapore: World Scientific, 2001.
- JIANG, Y.-Q.; ZHANG, P.; WONG, S.; LIU, R.-X. A higher-order macroscopic model for pedestrian flows. *Physica A: Statistical Mechanics and its Applications*, v. 389, n. 21, p. 4623–4635, 2010.
- KIRCHNER, A.; SCHADSCHNEIDER, A. Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. *Physica A: Statistical Mechanics and its Applications*, v. 312, n. 1-2, p. 260–276, 2002.
- LI, X.; KUANG, H.; FAN, Y. Lattice hydrodynamic model of pedestrian flow considering the asymmetric effect. *Communications in Nonlinear Science and Numerical Simulation*, v. 17, n. 3, p. 1258–1263, 2012.
- NOWAK, S.; SCHADSCHNEIDER, A. Quantitative analysis of pedestrian counterflow in a cellular automaton model. *Physical Review E*, American Physical Society, v. 85, p. 066128, 2012.
- PARISI, D. R.; DORSO, C. O. Morphological and dynamical aspects of the room evacuation process. *Physica A: Statistical Mechanics and its Applications*, v. 385, n. 1, p. 343–355, 2007.
- PELECHANO, N.; ALLBECK, J. M.; BADLER, N. I. Controlling individual agents in high-density crowd simulation. In: EUROGRAPHICS ASSOCIATION. *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*. [S.l.], 2007. p. 99–108.
- PEREIRA, L. A.; DUCZMAL, L. H.; CRUZ, F. R. B. Congested emergency evacuation of a population using a finite automata approach. *Safety Science*, v. 51, n. 1, p. 267–272, 2013.
- RINALDI, P. R.; DALPONTE, D. D.; VENERE, M. J.; CLAUSSE, A. Cellular automata algorithm for simulation of surface flows in large plains. *Simulation Modelling Practice & Theory*, v. 15, n. 3, p. 315–327, 2007.
- SCHADSCHNEIDER, A. Cellular automaton approach to pedestrian dynamics - Theory. In: *Pedestrian and Evacuation Dynamics*. [S.l.: s.n.], 2002. p. 75–86.
- SCHADSCHNEIDER, A.; KLINGSCH, W.; KLÜPFEL, H.; KRETZ, T.; ROGSCH, C.; SEYFRIED, A. Encyclopedia of complexity and systems science. In: _____. New York, NY: Springer New York, 2009. cap. Evacuation Dynamics: Empirical Results, Modeling and Applications, p. 3142–3176.

- SEYFRIED, A.; STEFFEN, B.; LIPPERT, T. Basics of modelling the pedestrian flow. *Physica A: Statistical Mechanics and its Applications*, v. 368, n. 1, p. 232–238, 2006.
- SMITH, A.; JAMES, C.; JONES, R.; LANGSTON, P.; LESTER, E.; DRURY, J. Modelling contra-flow in crowd dynamics DEM simulation. *Safety Science*, v. 47, n. 3, p. 395–404, 2009.
- SONG, W.; XU, X.; WANG, B.-H.; NI, S. Simulation of evacuation processes using a multi-grid model for pedestrian dynamics. *Physica A: Statistical Mechanics and its Applications*, v. 363, n. 2, p. 492–500, 2006.
- STEPANOV, A.; MACGREGOR SMITH, J. Multi-objective evacuation routing in transportation networks. *European Journal of Operational Research*, v. 198, n. 2, p. 435–446, 2009.
- SUMA, Y.; YANAGISAWA, D.; NISHINARI, K. Anticipation effect in pedestrian dynamics: Modeling and experiments. *Physica A: Statistical Mechanics and its Applications*, v. 391, n. 1-2, p. 248–263, 2012.
- TIAN, H.-H.; HE, H.-D.; WEI, Y.-F.; YU, X.; LU, W.-Z. Lattice hydrodynamic model with bidirectional pedestrian flow. *Physica A: Statistical Mechanics and its Applications*, v. 388, n. 14, p. 2895–2902, 2009.
- TOFFOLI, T. Cellular automata as an alternative to (rather than an approximation of) differential equations in modeling physics. *Physica D: Nonlinear Phenomena*, v. 10, n. 1, p. 117–127, 1984.
- TREGENZA, P. R. *The Design of Interior Circulation*. New York, NY: Van Nostrand Reinhold Company, 1976.
- VON NEUMANN, J. *Theory of Self-Reproducing Automata*. Champaign, IL: University of Illinois Press, 1966.
- WOENSEL, T. van; CRUZ, F. R. B. A stochastic approach to traffic congestion costs. *Computers & Operations Research*, v. 36, n. 6, p. 1731–1739, 2009.
- WOLFRAM, S. *Cellular Automata and Complexity*. Reading, MA: Addison-Wesley, 1994.
- WOLFRAM, S. *A New Kind of Science*. Champaign, IL: Wolfram Media, Inc., 2002.
- ZHENG, X.; LIU, M. Forecasting model for pedestrian distribution under emergency evacuation. *Reliability Engineering & System Safety*, v. 95, n. 11, p. 1186–1192, 2010.

APÊNDICE A - PROGRAMA DE SIMULAÇÃO

```

// Simulacao de fluxo de pessoas: nova versao
// 2016/05/27
#include <stdio.h>
#include <math.h>
#include <windows.h>
#define IA 16807
#define IM 2147483647
#define AM (1.0/IM)
#define IQ 127773
#define IR 2836
#define MASK 45
#include <GL/freeglut.h>
#include <stdlib.h>
long nx,ny, pixsize;
int aux1,aux2,marc,achou,d1,depth,q,pe,cont=0,cont2=0,cont3=0,conta,contb,contc,m,sentnovo1,sentnovo2,sentpr,
sentpopant;
long k2,n2,ng, contret,peoples,contp[10000],auxi,auxj,auxii,auxjj,mi,mj,auxk,trocak,u,u2,nsimsmax,nsims;
long g=0,replicacoes,i,j,lance,lancemax,lancelen,lancelenmax;
long aux,pause1,pause2;
int printflag,exibesimulacao,japassou1,japassou2,japassou3,r1,r2,saidaf,conter;
int somastatus,t1,t2,k,a,b1,b3,i00,j00,i000,j000,ntries;
char layout[200],trajetoria[200],trajetoria2[200],trajetoria3[200],volta;
long idum; //Semente da geracao de numeros aleatorios
double auxd,pEntra,temp,somatemp,somataux,somataux2,mataux[3][3],mataux2[5][5],
matw2[5][5],Ka,kd,ks,kr,kr2=1,probsm,rel,rel2, // pEntra=PInicial*pSai
//rel define a densidade na vizinhanca para que o sujeito mude de rota.
// se o sujeito segue rota contraria a vizinhanca, quanto maior o rel, menor sera a
densidade na vizinhanca para mudanca de rota.
//rel2 define a vizinhanca anterior para que o sujeito mude de rota
// quanto menor o rel2, maior sera a vizinhanca anterior, ou seja mais incluido
dentro de sua populacao ele podera estar quando decidir mudar de rota
//promd define a prob. de mudanca de rota em fila. Se ha uma ocupacao muito alta na
sua frente, a rota é mudada com probabilidade promd

```

```

//probsm define a probabilidade de mudança de rota quando há uma alta densidade de
//sujeitos em rota contrária.
//pe define a zivinhaça a ser considerada para mudança de rota.
gamma, alppa, DeltaD, escolha, total,
pSai, media, sp, sc1, sc2, pp1, pp2, //
pInicial; //
//Ka=5,kd=1, ks=10, promd=0.5,probsm=0.8,rel=3,rel2=3,gamma=0.1,alppa=0.3
// pe = vizinhança para mudança de potencial
// kr = probabilidade de mudar de potencial em congestionamento

double auxr;
int contel, faux, ocup[200][200],codigo[200][200],codigo1[200][200],codigo2[200][200],codigo3[200][200],CA
[200][200][9],tabn[200][200],tipo[200][200],traj[200][200],traj2[200][200],traj3[200][200],trajx[200][200],
dur[200][200],mark[200][200],auxw[200][200],auxtraj;
int ndepaux, celobjaux, situaux, codigaux, sort[40000],ndep[40000],sentpop[40000],situacao[40000],id[40000], sentaux, r3,
ni, nj, imin, imax, jmin, jmax, front[500], posfronti[500], posfrontj[500];
long tab[200][200];
long orderi[10000],orderj[10000],orderiant[10000],orderjant[10000],orderiantaux,orderjantaux, smat;
double ds1, ds2, mp[4],menord, maiord, D1[200][200],D2[200][200],D3[200][200], matw[3][3], mat3[3][3],mat5[5][5],mat7
[7][7],soma,acc3[9],acc5[25],r,S[200][200],aux3,dist[200][200],distt1[200][200],distt2[200][200],distt3
[200][200],distc1[200][200],distc2[200][200],distc3[200][200],distant[40000];
float rnd(long idum0);
double logpow( double x1, double x2 );
double power( double x1, double x2 );
double sqr( double x );
int mover( int i );
int movehor( int i );
int I( double x );
int mov( int range, int i0, int j0, int *ni, int *nj );
void plote( int x, int y, int t, int c );
void finner( int i000, int j000, int *i00, int *j00 );
void fbound( int i000, int j000, int *i00, int *j00 );
FILE *par;
FILE *sai;
FILE *sait;
FILE *leia;
FILE *leiadir;
FILE *leiacampo;
// Função callback chamada para fazer o desenho

int I( double x ) {
    if (x>0) return(1);
    else return(0);
}
int pos( int x ) {
    if (x>=0) return(x);
    else return((-1)*x);
}

```

```

void redefine (int pa,int pa2) {
    int t,i,j;
    int cont;
    int t1 , ET[26];
    soma = 0.0;
    smat=9;
    cont=0;
    for (t1=1; t1 <=25; t1++) ET[t1] = 0;
    for (i=-2; i <=2; i++)
        for (j=-2; j <=2; j++) {
            cont++;
            for (t1=1; t1 <=8; t1++) {
                if ((orderi[k]+j)>=0) if ((orderj[k]+i)>=0) ET[cont] = ET[cont]+ CA[orderi[k]+j][
                    orderj[k]+i][t1];
            }
        }
    cont=0;
    switch (pa) {
        case 1:
            if ( ( tab[orderi[k]+1][orderj[k]] + tab[orderi[k]+2][orderj[k]] ) > 0 ) {
                smat=9;
            } else {
                smat=25;
            }
            matw[0][0]=0.01;
            matw[1][0]=0.05;
            matw[2][0]=0.20;
            matw[0][1]=0.02;
            matw[1][1]=0.06;
            matw[2][1]=0.40;
            matw[0][2]=0.01;
            matw[1][2]=0.05;
            matw[2][2]=0.20;
            matw2[0][0]=0.00;
            matw2[1][0]=0.00;
            matw2[2][0]=0.00;
            matw2[3][0]=0.00;
            matw2[4][0]=0.00;
            matw2[0][1]=0.00;
            matw2[1][1]=0.03;
            matw2[2][1]=0.03;
            matw2[3][1]=0.05;
            matw2[4][1]=0.07;
            matw2[0][2]=0.00;
            matw2[1][2]=0.03;
            matw2[2][2]=0.10;
            matw2[3][2]=0.25;
            matw2[4][2]=0.26;
            matw2[0][3]=0.00;

```

```

matw2[1][3]=0.03;
matw2[2][3]=0.03;
matw2[3][3]=0.05;
matw2[4][3]=0.07;
matw2[0][4]=0.00;
matw2[1][4]=0.00;
matw2[2][4]=0.00;
matw2[3][4]=0.00;
matw2[4][4]=0.00;
for (i=-2; i<=2; i++)
    for (j=-2; j<=2; j++) {
        cont++;
        if ((orderi[k]+j)>=0) if ((orderj[k]+i)>=0) ET[cont] = ET[cont] - CA[orderi[
            k]+j][orderj[k]+i][1];
    }
break;
case 2:
if ( ( tab[orderi[k]][orderj[k]-1] + tab[orderi[k]][orderj[k]-2] ) > 0 ) {
    smat=9;
} else {
    smat=25;
}
matw[0][0]=0.20;
matw[1][0]=0.40;
matw[2][0]=0.20;
matw[0][1]=0.05;
matw[1][1]=0.06;
matw[2][1]=0.05;
matw[0][2]=0.01;
matw[1][2]=0.02;
matw[2][2]=0.01;
matw2[0][0]=0.00;
matw2[1][0]=0.07;
matw2[2][0]=0.26;
matw2[3][0]=0.07;
matw2[4][0]=0.00;
matw2[0][1]=0.00;
matw2[1][1]=0.05;
matw2[2][1]=0.25;
matw2[3][1]=0.05;
matw2[4][1]=0.00;
matw2[0][2]=0.00;
matw2[1][2]=0.03;
matw2[2][2]=0.10;
matw2[3][2]=0.03;
matw2[4][2]=0.00;
matw2[0][3]=0.00;
matw2[1][3]=0.03;
matw2[2][3]=0.03;

```

```

matw2[3][3]=0.03;
matw2[4][3]=0.00;
matw2[0][4]=0.00;
matw2[1][4]=0.00;
matw2[2][4]=0.00;
matw2[3][4]=0.00;
matw2[4][4]=0.00;
for (i=-2; i<=2; i++)
    for (j=-2; j<=2; j++) {
        cont++;
        if ((orderi[k]+j)>=0) if ((orderj[k]+i)>=0) ET[cont] = ET[cont] - CA[orderi[
            k]+j][orderj[k]+i][2];
    }
break;
case 3:
if ( ( tab[orderi[k]][orderj[k]+1] + tab[orderi[k]][orderj[k]+2]) > 0 ) {
    smat=9;
} else {
    smat=25;
}
matw[0][0]=0.01;
matw[1][0]=0.02;
matw[2][0]=0.01;
matw[0][1]=0.05;
matw[1][1]=0.06;
matw[2][1]=0.05;
matw[0][2]=0.20;
matw[1][2]=0.40;
matw[2][2]=0.20;
matw2[0][0]=0.00;
matw2[1][0]=0.00;
matw2[2][0]=0.00;
matw2[3][0]=0.00;
matw2[4][0]=0.00;
matw2[0][1]=0.00;
matw2[1][1]=0.03;
matw2[2][1]=0.03;
matw2[3][1]=0.03;
matw2[4][1]=0.00;
matw2[0][2]=0.00;
matw2[1][2]=0.03;
matw2[2][2]=0.10;
matw2[3][2]=0.03;
matw2[4][2]=0.00;
matw2[0][3]=0.00;
matw2[1][3]=0.05;
matw2[2][3]=0.25;
matw2[3][3]=0.05;
matw2[4][3]=0.00;

```

```

matw2[0][4]=0.00;
matw2[1][4]=0.07;
matw2[2][4]=0.26;
matw2[3][4]=0.07;
matw2[4][4]=0.00;
for (i=-2; i<=2; i++)
    for (j=-2; j<=2; j++) {
        cont++;
        if ((orderi[k]+j)>=0) if ((orderj[k]+i)>=0) ET[cont] = ET[cont] - CA[orderi[
            k]+j][orderj[k]+i][4];
    }
break;
case 4:
if ( ( tab[orderi[k]-1][orderj[k]] + tab[orderi[k]-2][orderj[k]] ) > 0 ) {
    smat=9;
} else {
    smat=25;
}
matw[0][0]=0.20;
matw[1][0]=0.05;
matw[2][0]=0.01;
matw[0][1]=0.40;
matw[1][1]=0.06;
matw[2][1]=0.02;
matw[0][2]=0.20;
matw[1][2]=0.05;
matw[2][2]=0.01;
matw2[0][0]=0.00;
matw2[1][0]=0.00;
matw2[2][0]=0.00;
matw2[3][0]=0.00;
matw2[4][0]=0.00;
matw2[0][1]=0.07;
matw2[1][1]=0.05;
matw2[2][1]=0.03;
matw2[3][1]=0.03;
matw2[4][1]=0.00;
matw2[0][2]=0.26;
matw2[1][2]=0.25;
matw2[2][2]=0.10;
matw2[3][2]=0.03;
matw2[4][2]=0.00;
matw2[0][3]=0.07;
matw2[1][3]=0.05;
matw2[2][3]=0.03;
matw2[3][3]=0.03;
matw2[4][3]=0.00;
matw2[0][4]=0.00;
matw2[1][4]=0.00;

```



```

matw2[2][4]=0.00;
matw2[3][4]=0.00;
matw2[4][4]=0.00;
for (i=-2; i<=2; i++)
    for (j=-2; j<=2; j++) {
        cont++;
        if ((orderi[k]+j)>=0) if ((orderj[k]+i)>=0) ET[cont] = ET[cont] - CA[orderi[
            k]+j][orderj[k]+i][3];
    }
break;
case 5:
if ( ( tab[orderi[k]+1][orderj[k]-1] + tab[orderi[k]+2][orderj[k]-2]) > 0 ) {
    smat=9;
} else {
    smat=25;
}
matw[0][0]=0.05;
matw[1][0]=0.20;
matw[2][0]=0.40;
matw[0][1]=0.01;
matw[1][1]=0.06;
matw[2][1]=0.20;
matw[0][2]=0.02;
matw[1][2]=0.01;
matw[2][2]=0.05;
matw2[0][0]=0.00;
matw2[1][0]=0.00;
matw2[2][0]=0.00;
matw2[3][0]=0.07;
matw2[4][0]=0.26;
matw2[0][1]=0.00;
matw2[1][1]=0.03;
matw2[2][1]=0.05;
matw2[3][1]=0.25;
matw2[4][1]=0.07;
matw2[0][2]=0.00;
matw2[1][2]=0.03;
matw2[2][2]=0.10;
matw2[3][2]=0.05;
matw2[4][2]=0.00;
matw2[0][3]=0.00;
matw2[1][3]=0.03;
matw2[2][3]=0.03;
matw2[3][3]=0.03;
matw2[4][3]=0.00;
matw2[0][4]=0.00;
matw2[1][4]=0.00;
matw2[2][4]=0.00;
matw2[3][4]=0.00;

```

```

matw2[4][4]=0.00;
for (i=-2; i<=2; i++)
    for (j=-2; j<=2; j++) {
        cont++;
        if ((orderi[k]+j)>=0) if ((orderj[k]+i)>=0) ET[cont] = ET[cont] - CA[orderi[
            k]+j][orderj[k]+i][5];
    }
break ;

```

case 6:

```

if ( ( tab[orderi[k]+1][orderj[k]+1] + tab[orderi[k]+2][orderj[k]+2]) > 0 ) {
    smat=9;
} else {
    smat=25;
}

matw[0][0]=0.02;
matw[1][0]=0.01;
matw[2][0]=0.05;
matw[0][1]=0.01;
matw[1][1]=0.06;
matw[2][1]=0.20;
matw[0][2]=0.05;
matw[1][2]=0.20;
matw[2][2]=0.40;
matw2[0][0]=0.00;
matw2[1][0]=0.00;
matw2[2][0]=0.00;
matw2[3][0]=0.00;
matw2[4][0]=0.00;
matw2[0][1]=0.00;
matw2[1][1]=0.03;
matw2[2][1]=0.03;
matw2[3][1]=0.03;
matw2[4][1]=0.00;
matw2[0][2]=0.00;
matw2[1][2]=0.03;
matw2[2][2]=0.10;
matw2[3][2]=0.05;
matw2[4][2]=0.00;
matw2[0][3]=0.00;
matw2[1][3]=0.03;
matw2[2][3]=0.05;
matw2[3][3]=0.25;
matw2[4][3]=0.07;
matw2[0][4]=0.00;
matw2[1][4]=0.00;
matw2[2][4]=0.00;
matw2[3][4]=0.07;
matw2[4][4]=0.26;

```

```

for (i=-2; i<=2; i++)
    for (j=-2; j<=2; j++) {
        cont++;
        if ((orderi[k]+j)>=0) if ((orderj[k]+i)>=0) ET[cont] = ET[cont] - CA[orderi[
            k]+j][orderj[k]+i][6];
    }
break ;
case 7:
if ( ( tab[orderi[k]-1][orderj[k]-1] + tab[orderi[k]-2][orderj[k]-2]) > 0 ) {
    smat=9;
} else {
    smat=25;
}
matw[0][0]=0.40;
matw[1][0]=0.20;
matw[2][0]=0.05;
matw[0][1]=0.20;
matw[1][1]=0.06;
matw[2][1]=0.01;
matw[0][2]=0.05;
matw[1][2]=0.01;
matw[2][2]=0.02;
matw2[0][0]=0.26;
matw2[1][0]=0.07;
matw2[2][0]=0.00;
matw2[3][0]=0.00;
matw2[4][0]=0.00;
matw2[0][1]=0.07;
matw2[1][1]=0.25;
matw2[2][1]=0.05;
matw2[3][1]=0.03;
matw2[4][1]=0.00;
matw2[0][2]=0.00;
matw2[1][2]=0.05;
matw2[2][2]=0.10;
matw2[3][2]=0.03;
matw2[4][2]=0.00;
matw2[0][3]=0.00;
matw2[1][3]=0.03;
matw2[2][3]=0.03;
matw2[3][3]=0.03;
matw2[4][3]=0.00;
matw2[0][4]=0.00;
matw2[1][4]=0.00;
matw2[2][4]=0.00;
matw2[3][4]=0.00;
matw2[4][4]=0.00;
for (i=-2; i<=2; i++)
    for (j=-2; j<=2; j++) {

```

```

        cont++;
        if ((orderi[k]+j)>=0) if ((orderj[k]+i)>=0) ET[cont] = ET[cont] - CA[orderi[
            k]+j][orderj[k]+i][7];
    }
    break ;
case 8:

    if ( ( tab[orderi[k]-1][orderj[k]+1] + tab[orderi[k]-2][orderj[k]+2]) > 0 ) {
        smat=9;
    } else {
        smat=25;
    }
    matw[0][0]=0.05;
    matw[1][0]=0.01;
    matw[2][0]=0.02;
    matw[0][1]=0.20;
    matw[1][1]=0.06;
    matw[2][1]=0.01;
    matw[0][2]=0.40;
    matw[1][2]=0.20;
    matw[2][2]=0.05;
    matw2[0][0]=0.00;
    matw2[1][0]=0.00;
    matw2[2][0]=0.00;
    matw2[3][0]=0.00;
    matw2[4][0]=0.00;
    matw2[0][1]=0.00;
    matw2[1][1]=0.03;
    matw2[2][1]=0.03;
    matw2[3][1]=0.03;
    matw2[4][1]=0.00;
    matw2[0][2]=0.00;
    matw2[1][2]=0.05;
    matw2[2][2]=0.10;
    matw2[3][2]=0.03;
    matw2[4][2]=0.00;
    matw2[0][3]=0.07;
    matw2[1][3]=0.25;
    matw2[2][3]=0.05;
    matw2[3][3]=0.03;
    matw2[4][3]=0.00;
    matw2[0][4]=0.26;
    matw2[1][4]=0.07;
    matw2[2][4]=0.00;
    matw2[3][4]=0.00;
    matw2[4][4]=0.00;
    for (i=-2; i<=2; i++)
        for (j=-2; j<=2; j++) {
            cont++;

```

```

        if ((orderi[k]+j)>=0) if ((orderj[k]+i)>=0) ET[cont] = ET[cont] - CA[orderi[
            k]+j][orderj[k]+i][8];
    }
    break;
}
if (pa2>=3) smat=9;
if (smat==9) {
    somataux = 0.0;
    if (pa2==1) {
        mataux[0][0]= I(matw[0][0])* exp( ks* matw[0][0] + kd*D1[ orderi [k]-1][ orderj [k]-1]-Ka*ET[7])
            ; /* exp(ks*S[ orderi [k-1]][ orderj [k-1]]);
        mataux[0][1]= I(matw[0][1])* exp( ks* matw[0][1] + kd*D1[ orderi [k]][ orderj [k]-1]-Ka*ET[8]);
            /* exp(ks*S[ orderi [k]][ orderj [k-1]]);
        mataux[0][2]= I(matw[0][2])* exp( ks* matw[0][2] + kd*D1[ orderi [k]+1][ orderj [k]-1]-Ka*ET[9])
            ; /* * exp(ks*S[ orderi [k+1]][ orderj [k-1]]);
        mataux[1][0]= I(matw[1][0])* exp( ks* matw[1][0] + kd*D1[ orderi [k]-1][ orderj [k]-Ka*ET[12]);
            /* exp(ks*S[ orderi [k-1]][ orderj [k]]);
        mataux[1][1]= I(matw[1][1])* exp( ks* matw[1][1] + kd*D1[ orderi [k]][ orderj [k]-Ka*ET[13]);
            /* exp(ks*S[ orderi [k]][ orderj [k]]);
        mataux[1][2]= I(matw[1][2])* exp( ks* matw[1][2] + kd*D1[ orderi [k]+1][ orderj [k]-Ka*ET[14]);
            /* exp(ks*S[ orderi [k+1]][ orderj [k]]);
        mataux[2][0]= I(matw[2][0])* exp( ks* matw[2][0] + kd*D1[ orderi [k]-1][ orderj [k]+1]-Ka*ET
            [17]); /* exp(ks*S[ orderi [k-1]][ orderj [k+1]]);
        mataux[2][1]= I(matw[2][1])* exp( ks* matw[2][1] + kd*D1[ orderi [k]][ orderj [k]+1]-Ka*ET[18]);
            /* exp(ks*S[ orderi [k]][ orderj [k+1]]);
        mataux[2][2]= I(matw[2][2])* exp( ks* matw[2][2] + kd*D1[ orderi [k]+1][ orderj [k]+1]-Ka*ET
            [19]); /* exp(ks*S[ orderi [k+1]][ orderj [k+1]]);
    } else if (pa2==2) {
        mataux[0][0]= I(matw[0][0])* exp( ks* matw[0][0] + kd*D2[ orderi [k]-1][ orderj [k]-1]-Ka*ET[7])
            ; /* exp(ks*S[ orderi [k-1]][ orderj [k-1]]);
        mataux[0][1]= I(matw[0][1])* exp( ks* matw[0][1] + kd*D2[ orderi [k]][ orderj [k]-1]-Ka*ET[8]);
            /* exp(ks*S[ orderi [k]][ orderj [k-1]]);
        mataux[0][2]= I(matw[0][2])* exp( ks* matw[0][2] + kd*D2[ orderi [k]+1][ orderj [k]-1]-Ka*ET[9])
            ; /* * exp(ks*S[ orderi [k+1]][ orderj [k-1]]);
        mataux[1][0]= I(matw[1][0])* exp( ks* matw[1][0] + kd*D2[ orderi [k]-1][ orderj [k]-Ka*ET[12]);
            /* exp(ks*S[ orderi [k-1]][ orderj [k]]);
        mataux[1][1]= I(matw[1][1])* exp( ks* matw[1][1] + kd*D2[ orderi [k]][ orderj [k]-Ka*ET[13]);
            /* exp(ks*S[ orderi [k]][ orderj [k]]);
        mataux[1][2]= I(matw[1][2])* exp( ks* matw[1][2] + kd*D2[ orderi [k]+1][ orderj [k]-Ka*ET[14]);
            /* exp(ks*S[ orderi [k+1]][ orderj [k]]);
        mataux[2][0]= I(matw[2][0])* exp( ks* matw[2][0] + kd*D2[ orderi [k]-1][ orderj [k]+1]-Ka*ET
            [17]); /* exp(ks*S[ orderi [k-1]][ orderj [k+1]]);
        mataux[2][1]= I(matw[2][1])* exp( ks* matw[2][1] + kd*D2[ orderi [k]][ orderj [k]+1]-Ka*ET[18]);
            /* exp(ks*S[ orderi [k]][ orderj [k+1]]);
        mataux[2][2]= I(matw[2][2])* exp( ks* matw[2][2] + kd*D2[ orderi [k]+1][ orderj [k]+1]-Ka*ET
            [19]); /* exp(ks*S[ orderi [k+1]][ orderj [k+1]]);
    } else if (pa2==3) {
        mataux[0][0]= I(matw[0][0])* exp( ks* matw[0][0]+ kd*D3[ orderi [k]-1][ orderj [k]-1] - Ka*ET
            [7]); /* exp(ks*S[ orderi [k-1]][ orderj [k-1]]);

```

```

mataux[0][1]= I(matw[0][1])* exp( ks* matw[0][1]+ kd*D3[ orderi [k]][ orderj [k]-1] - Ka*ET[8]);
    /* exp(ks*S[ orderi [k]][ orderj [k-1]]);
mataux[0][2]= I(matw[0][2])* exp( ks* matw[0][2]+ kd*D3[ orderi [k]+1][ orderj [k]-1]- Ka*ET[9])
    ; /* exp(ks*S[ orderi [k+1]][ orderj [k-1]]);
mataux[1][0]= I(matw[1][0])* exp( ks* matw[1][0]+ kd*D3[ orderi [k]-1][ orderj [k]] - Ka*ET[12])
    ; /* exp(ks*S[ orderi [k-1]][ orderj [k]]);
mataux[1][1]= I(matw[1][1])* exp( ks* matw[1][1]+ kd*D3[ orderi [k]][ orderj [k]] - Ka*ET[13]);
    /* exp(ks*S[ orderi [k]][ orderj [k]]);
mataux[1][2]= I(matw[1][2])* exp( ks* matw[1][2]+ kd*D3[ orderi [k]+1][ orderj [k]] - Ka*ET[14])
    ; /* exp(ks*S[ orderi [k+1]][ orderj [k]]);
mataux[2][0]= I(matw[2][0])* exp( ks* matw[2][0]+ kd*D3[ orderi [k]-1][ orderj [k]+1] - Ka*ET
    [17]); /* exp(ks*S[ orderi [k-1]][ orderj [k+1]]);
mataux[2][1]= I(matw[2][1])* exp( ks* matw[2][1]+ kd*D3[ orderi [k]][ orderj [k]+1] - Ka*ET[18])
    ; /* exp(ks*S[ orderi [k]][ orderj [k+1]]);
mataux[2][2]= I(matw[2][2])* exp( ks* matw[2][2]+ kd*D3[ orderi [k]+1][ orderj [k]+1] - Ka*ET
    [19]); /* exp(ks*S[ orderi [k+1]][ orderj [k+1]]);
} // else {mataux[0][0]=mataux[0][1]=mataux[0][2]=mataux[1][0]=mataux[1][2]=mataux[2][0]=mataux
    [2][1]=mataux[2][2]=1; mataux[1][1]=30;}
/* mataux[0][0]= exp(kd*D[ orderi [k-1]][ orderj [k-1]])* exp(ks*dist [ orderi [k-1]][ orderj [k-1]]);
mataux[0][1]= exp(kd*D[ orderi [k]][ orderj [k-1]])* exp(ks*dist [ orderi [k]][ orderj [k-1]]);
mataux[0][2]= exp(kd*D[ orderi [k+1]][ orderj [k-1]]) * exp(ks*dist [ orderi [k+1]][ orderj [k-1]]);
mataux[1][0]= exp(kd*D[ orderi [k-1]][ orderj [k]]) * exp(ks*dist [ orderi [k-1]][ orderj [k]]);
mataux[1][1]= exp(kd*D[ orderi [k]][ orderj [k]]) * exp(ks*dist [ orderi [k]][ orderj [k]]);
mataux[1][2]= exp(kd*D[ orderi [k+1]][ orderj [k]]) * exp(ks*dist [ orderi [k+1]][ orderj [k]]);
mataux[2][0]= exp(kd*D[ orderi [k-1]][ orderj [k+1]]) * exp(ks*dist [ orderi [k-1]][ orderj [k+1]]);
mataux[2][1]= exp(kd*D[ orderi [k]][ orderj [k+1]]) * exp(ks*dist [ orderi [k]][ orderj [k+1]]);
mataux[2][2]= exp(kd*D[ orderi [k+1]][ orderj [k+1]]) * exp(ks*dist [ orderi [k+1]][ orderj [k+1]]); */
} else {
somataux2= 0.0;
if(pa2==1) {
mataux2[0][0]= I(matw2[0][0])* exp( ks* matw2[0][0]+ kd*D1[ orderi [k]-2][ orderj [k]-2]-Ka*ET
    [1]); /* exp(ks*dist [ orderi [k-2]][ orderj [k-2]]);
mataux2[0][1]= I(matw2[0][1])* exp( ks* matw2[0][1]+ kd*D1[ orderi [k]-1][ orderj [k]-2]-Ka*ET
    [2]); /* exp(ks*dist [ orderi [k-1]][ orderj [k-2]]); //
mataux2[0][2]= I(matw2[0][2])* exp( ks* matw2[0][2]+ kd*D1[ orderi [k]][ orderj [k]-2]-Ka*ET[3])
    ; /* exp(ks*dist [ orderi [k]][ orderj [k-2]]);
mataux2[0][3]= I(matw2[0][3])* exp( ks* matw2[0][3]+ kd*D1[ orderi [k]+1][ orderj [k]-2]-Ka*ET
    [4]); /* exp(ks*dist [ orderi [k+1]][ orderj [k-2]]);
mataux2[0][4]= I(matw2[0][4])* exp( ks* matw2[0][4]+ kd*D1[ orderi [k]+2][ orderj [k]-2]-Ka*ET
    [5]); /* exp(ks*dist [ orderi [k+2]][ orderj [k-2]]);
mataux2[1][0]= I(matw2[1][0])* exp( ks* matw2[1][0]+ kd*D1[ orderi [k]-2][ orderj [k]-1]-Ka*ET
    [6]); /* exp(ks*dist [ orderi [k-2]][ orderj [k-1]]);
mataux2[1][1]= I(matw2[1][1])* exp( ks* matw2[1][1]+ kd*D1[ orderi [k]-1][ orderj [k]-1]-Ka*ET
    [7]); /* exp(ks*dist [ orderi [k-1]][ orderj [k-1]]);
mataux2[1][2]= I(matw2[1][2])* exp( ks* matw2[1][2]+ kd*D1[ orderi [k]][ orderj [k]-1]-Ka*ET[8])
    ; /* exp(ks*dist [ orderi [k]][ orderj [k-1]]);
mataux2[1][3]= I(matw2[1][3])* exp( ks* matw2[1][3]+ kd*D1[ orderi [k]+1][ orderj [k]-1]-Ka*ET
    [9]); /* exp(ks*dist [ orderi [k+1]][ orderj [k-1]]);
mataux2[1][4]= I(matw2[1][4])* exp( ks* matw2[1][4]+ kd*D1[ orderi [k]+2][ orderj [k]-1]-Ka*ET

```

```

[10]); /* exp(ks*dist[orderi[k+2]][orderj[k-1]]);
mataux2[2][0]= I(matw2[2][0])* exp( ks* matw2[2][0]+ kd*D1[orderi[k]-2][orderj[k]]-Ka*ET
[11]); /* exp(ks*dist[orderi[k-2]][orderj[k]]);
mataux2[2][1]= I(matw2[2][1])* exp( ks* matw2[2][1]+ kd*D1[orderi[k]-1][orderj[k]]-Ka*ET
[12]); /* exp(ks*dist[orderi[k-1]][orderj[k]]); //
mataux2[2][2]= I(matw2[2][2])* exp( ks* matw2[2][2]+ kd*D1[orderi[k]][orderj[k]]-Ka*ET[13]);
/* exp(ks*dist[orderi[k]][orderj[k]]);
mataux2[2][3]= I(matw2[2][3])* exp( ks* matw2[2][3]+ kd*D1[orderi[k]+1][orderj[k]]-Ka*ET
[14]); /* exp(ks*dist[orderi[k+1]][orderj[k]]);
mataux2[2][4]= I(matw2[2][4])* exp( ks* matw2[2][4]+ kd*D1[orderi[k]+2][orderj[k]]-Ka*ET
[15]); /* exp(ks*dist[orderi[k+2]][orderj[k]]);
mataux2[3][0]= I(matw2[3][0])* exp( ks* matw2[3][0]+ kd*D1[orderi[k]-2][orderj[k+1]]-Ka*ET
[16]); /* exp(ks*dist[orderi[k-2]][orderj[k+1]]); //
mataux2[3][1]= I(matw2[3][1])* exp( ks* matw2[3][1]+ kd*D1[orderi[k]-1][orderj[k+1]]-Ka*ET
[17]); /* exp(ks*dist[orderi[k-1]][orderj[k+1]]);
mataux2[3][2]= I(matw2[3][2])* exp( ks* matw2[3][2]+ kd*D1[orderi[k]][orderj[k+1]]-Ka*ET
[18]); /* exp(ks*dist[orderi[k]][orderj[k+1]]);
mataux2[3][3]= I(matw2[3][3])* exp( ks* matw2[3][3]+ kd*D1[orderi[k]+1][orderj[k+1]]-Ka*ET
[19]); /* exp(ks*dist[orderi[k+1]][orderj[k+1]]);
mataux2[3][4]= I(matw2[3][4])* exp( ks* matw2[3][4]+ kd*D1[orderi[k]+2][orderj[k+1]]-Ka*ET
[20]); /* exp(ks*dist[orderi[k+2]][orderj[k+1]]); //
mataux2[4][0]= I(matw2[4][0])* exp( ks* matw2[4][0]+ kd*D1[orderi[k]-2][orderj[k+2]]-Ka*ET
[21]); /* exp(ks*dist[orderi[k-2]][orderj[k+2]]);
mataux2[4][1]= I(matw2[4][1])* exp( ks* matw2[4][1]+ kd*D1[orderi[k]-1][orderj[k+2]]-Ka*ET
[22]); /* exp(ks*dist[orderi[k-1]][orderj[k+2]]);
mataux2[4][2]= I(matw2[4][2])* exp( ks* matw2[4][2]+ kd*D1[orderi[k]][orderj[k+2]]-Ka*ET
[23]); /* exp(ks*dist[orderi[k]][orderj[k+2]]);
mataux2[4][3]= I(matw2[4][3])* exp( ks* matw2[4][3]+ kd*D1[orderi[k]+1][orderj[k+2]]-Ka*ET
[24]); /* exp(ks*dist[orderi[k+1]][orderj[k+2]]); //
mataux2[4][4]= I(matw2[4][4])* exp( ks* matw2[4][4]+ kd*D1[orderi[k]+2][orderj[k+2]]-Ka*ET
[25]); /* exp(ks*dist[orderi[k+2]][orderj[k+2]]);
} else if (pa2==2) {
mataux2[0][0]= I(matw2[0][0])* exp( ks* matw2[0][0]+ kd*D2[orderi[k]-2][orderj[k]-2]-Ka*ET
[1]); /* exp(ks*dist[orderi[k-2]][orderj[k-2]]);
mataux2[0][1]= I(matw2[0][1])* exp( ks* matw2[0][1]+ kd*D2[orderi[k]-1][orderj[k]-2]-Ka*ET
[2]); /* exp(ks*dist[orderi[k-1]][orderj[k-2]]); //
mataux2[0][2]= I(matw2[0][2])* exp( ks* matw2[0][2]+ kd*D2[orderi[k]][orderj[k]-2]-Ka*ET[3]
; /* exp(ks*dist[orderi[k]][orderj[k-2]]);
mataux2[0][3]= I(matw2[0][3])* exp( ks* matw2[0][3]+ kd*D2[orderi[k]+1][orderj[k]-2]-Ka*ET
[4]); /* exp(ks*dist[orderi[k+1]][orderj[k-2]]);
mataux2[0][4]= I(matw2[0][4])* exp( ks* matw2[0][4]+ kd*D2[orderi[k]+2][orderj[k]-2]-Ka*ET
[5]); /* exp(ks*dist[orderi[k+2]][orderj[k-2]]);
mataux2[1][0]= I(matw2[1][0])* exp( ks* matw2[1][0]+ kd*D2[orderi[k]-2][orderj[k]-1]-Ka*ET
[6]); /* exp(ks*dist[orderi[k-2]][orderj[k-1]]);
mataux2[1][1]= I(matw2[1][1])* exp( ks* matw2[1][1]+ kd*D2[orderi[k]-1][orderj[k]-1]-Ka*ET
[7]); /* exp(ks*dist[orderi[k-1]][orderj[k-1]]);
mataux2[1][2]= I(matw2[1][2])* exp( ks* matw2[1][2]+ kd*D2[orderi[k]][orderj[k]-1]-Ka*ET[8]
; /* exp(ks*dist[orderi[k]][orderj[k-1]]);
mataux2[1][3]= I(matw2[1][3])* exp( ks* matw2[1][3]+ kd*D2[orderi[k]+1][orderj[k]-1]-Ka*ET

```

```

[9]); // * exp(ks*dist[orderi[k+1]][orderj[k-1]]);
mataux2[1][4]= I(matw2[1][4])* exp( ks* matw2[1][4]+ kd*D2[orderi[k]+2][orderj[k]-1]-Ka*ET
[10]); // * exp(ks*dist[orderi[k+2]][orderj[k-1]]);
mataux2[2][0]= I(matw2[2][0])* exp( ks* matw2[2][0]+ kd*D2[orderi[k]-2][orderj[k]]-Ka*ET
[11]); // * exp(ks*dist[orderi[k-2]][orderj[k]]);
mataux2[2][1]= I(matw2[2][1])* exp( ks* matw2[2][1]+ kd*D2[orderi[k]-1][orderj[k]]-Ka*ET
[12]); // * exp(ks*dist[orderi[k-1]][orderj[k]]); //
mataux2[2][2]= I(matw2[2][2])* exp( ks* matw2[2][2]+ kd*D2[orderi[k]][orderj[k]]-Ka*ET[13]);
// * exp(ks*dist[orderi[k]][orderj[k]]);
mataux2[2][3]= I(matw2[2][3])* exp( ks* matw2[2][3]+ kd*D2[orderi[k]+1][orderj[k]]-Ka*ET
[14]); // * exp(ks*dist[orderi[k+1]][orderj[k]]);
mataux2[2][4]= I(matw2[2][4])* exp( ks* matw2[2][4]+ kd*D2[orderi[k]+2][orderj[k]]-Ka*ET
[15]); // * exp(ks*dist[orderi[k+2]][orderj[k]]);
mataux2[3][0]= I(matw2[3][0])* exp( ks* matw2[3][0]+ kd*D2[orderi[k]-2][orderj[k+1]]-Ka*ET
[16]); // * exp(ks*dist[orderi[k-2]][orderj[k+1]]); //
mataux2[3][1]= I(matw2[3][1])* exp( ks* matw2[3][1]+ kd*D2[orderi[k]-1][orderj[k+1]]-Ka*ET
[17]); // * exp(ks*dist[orderi[k-1]][orderj[k+1]]);
mataux2[3][2]= I(matw2[3][2])* exp( ks* matw2[3][2]+ kd*D2[orderi[k]][orderj[k+1]]-Ka*ET
[18]); // * exp(ks*dist[orderi[k]][orderj[k+1]]);
mataux2[3][3]= I(matw2[3][3])* exp( ks* matw2[3][3]+ kd*D2[orderi[k]+1][orderj[k+1]]-Ka*ET
[19]); // * exp(ks*dist[orderi[k+1]][orderj[k+1]]);
mataux2[3][4]= I(matw2[3][4])* exp( ks* matw2[3][4]+ kd*D2[orderi[k]+2][orderj[k+1]]-Ka*ET
[20]); // * exp(ks*dist[orderi[k+2]][orderj[k+1]]); //
mataux2[4][0]= I(matw2[4][0])* exp( ks* matw2[4][0]+ kd*D2[orderi[k]-2][orderj[k+2]]-Ka*ET
[21]); // * exp(ks*dist[orderi[k-2]][orderj[k+2]]);
mataux2[4][1]= I(matw2[4][1])* exp( ks* matw2[4][1]+ kd*D2[orderi[k]-1][orderj[k+2]]-Ka*ET
[22]); // * exp(ks*dist[orderi[k-1]][orderj[k+2]]);
mataux2[4][2]= I(matw2[4][2])* exp( ks* matw2[4][2]+ kd*D2[orderi[k]][orderj[k+2]]-Ka*ET
[23]); // * exp(ks*dist[orderi[k]][orderj[k+2]]);
mataux2[4][3]= I(matw2[4][3])* exp( ks* matw2[4][3]+ kd*D2[orderi[k]+1][orderj[k+2]]-Ka*ET
[24]); // * exp(ks*dist[orderi[k+1]][orderj[k+2]]); //
mataux2[4][4]= I(matw2[4][4])* exp( ks* matw2[4][4]+ kd*D2[orderi[k]+2][orderj[k+2]]-Ka*ET
[25]); // * exp(ks*dist[o
} else if (pa2==3) {
mataux2[0][0]= I(matw2[0][0])* exp( ks* matw2[0][0]+ kd*D3[orderi[k]-2][orderj[k]-2]-Ka*ET
[1]); // * exp(ks*dist[orderi[k-2]][orderj[k-2]]);
mataux2[0][1]= I(matw2[0][1])* exp( ks* matw2[0][1]+ kd*D3[orderi[k]-1][orderj[k]-2]-Ka*ET
[2]); // * exp(ks*dist[orderi[k-1]][orderj[k-2]]); //
mataux2[0][2]= I(matw2[0][2])* exp( ks* matw2[0][2]+ kd*D3[orderi[k]][orderj[k]-2]-Ka*ET[3]
; // * exp(ks*dist[orderi[k]][orderj[k-2]]);
mataux2[0][3]= I(matw2[0][3])* exp( ks* matw2[0][3]+ kd*D3[orderi[k]+1][orderj[k]-2]-Ka*ET
[4]); // * exp(ks*dist[orderi[k+1]][orderj[k-2]]);
mataux2[0][4]= I(matw2[0][4])* exp( ks* matw2[0][4]+ kd*D3[orderi[k]+2][orderj[k]-2]-Ka*ET
[5]); // * exp(ks*dist[orderi[k+2]][orderj[k-2]]);
mataux2[1][0]= I(matw2[1][0])* exp( ks* matw2[1][0]+ kd*D3[orderi[k]-2][orderj[k]-1]-Ka*ET
[6]); // * exp(ks*dist[orderi[k-2]][orderj[k-1]]);
mataux2[1][1]= I(matw2[1][1])* exp( ks* matw2[1][1]+ kd*D3[orderi[k]-1][orderj[k]-1]-Ka*ET
[7]); // * exp(ks*dist[orderi[k-1]][orderj[k-1]]);
mataux2[1][2]= I(matw2[1][2])* exp( ks* matw2[1][2]+ kd*D3[orderi[k]][orderj[k]-1]-Ka*ET[8])

```



```

        :// * exp(ks*dist [orderi [k]][ orderj [k-1]]);
mataux2[1][3]= I(matw2[1][3])* exp( ks* matw2[1][3]+ kd*D3[orderi [k]+1][ orderj [k]-1]-Ka*ET
[9]); // * exp(ks*dist [orderi [k+1]][ orderj [k-1]]);
mataux2[1][4]= I(matw2[1][4])* exp( ks* matw2[1][4]+ kd*D3[orderi [k]+2][ orderj [k]-1]-Ka*ET
[10]); // * exp(ks*dist [orderi [k+2]][ orderj [k-1]]);
mataux2[2][0]= I(matw2[2][0])* exp( ks* matw2[2][0]+ kd*D3[orderi [k]-2][ orderj [k]]-Ka*ET
[11]); // * exp(ks*dist [orderi [k-2]][ orderj [k]]);
mataux2[2][1]= I(matw2[2][1])* exp( ks* matw2[2][1]+ kd*D3[orderi [k]-1][ orderj [k]]-Ka*ET
[12]); // * exp(ks*dist [orderi [k-1]][ orderj [k]]); //
mataux2[2][2]= I(matw2[2][2])* exp( ks* matw2[2][2]+ kd*D3[orderi [k]][ orderj [k]]-Ka*ET [13]);
// * exp(ks*dist [orderi [k]][ orderj [k]]);
mataux2[2][3]= I(matw2[2][3])* exp( ks* matw2[2][3]+ kd*D3[orderi [k]+1][ orderj [k]]-Ka*ET
[14]); // * exp(ks*dist [orderi [k+1]][ orderj [k]]);
mataux2[2][4]= I(matw2[2][4])* exp( ks* matw2[2][4]+ kd*D3[orderi [k]+2][ orderj [k]]-Ka*ET
[15]); // * exp(ks*dist [orderi [k+2]][ orderj [k]]);
mataux2[3][0]= I(matw2[3][0])* exp( ks* matw2[3][0]+ kd*D3[orderi [k]-2][ orderj [k+1]]-Ka*ET
[16]); // * exp(ks*dist [orderi [k-2]][ orderj [k+1]]); //
mataux2[3][1]= I(matw2[3][1])* exp( ks* matw2[3][1]+ kd*D3[orderi [k]-1][ orderj [k+1]]-Ka*ET
[17]); // * exp(ks*dist [orderi [k-1]][ orderj [k+1]]);
mataux2[3][2]= I(matw2[3][2])* exp( ks* matw2[3][2]+ kd*D3[orderi [k]][ orderj [k+1]]-Ka*ET
[18]); // * exp(ks*dist [orderi [k]][ orderj [k+1]]);
mataux2[3][3]= I(matw2[3][3])* exp( ks* matw2[3][3]+ kd*D3[orderi [k]+1][ orderj [k+1]]-Ka*ET
[19]); // * exp(ks*dist [orderi [k+1]][ orderj [k+1]]);
mataux2[3][4]= I(matw2[3][4])* exp( ks* matw2[3][4]+ kd*D3[orderi [k]+2][ orderj [k+1]]-Ka*ET
[20]); // * exp(ks*dist [orderi [k+2]][ orderj [k+1]]); //
mataux2[4][0]= I(matw2[4][0])* exp( ks* matw2[4][0]+ kd*D3[orderi [k]-2][ orderj [k+2]]-Ka*ET
[21]); // * exp(ks*dist [orderi [k-2]][ orderj [k+2]]);
mataux2[4][1]= I(matw2[4][1])* exp( ks* matw2[4][1]+ kd*D3[orderi [k]-1][ orderj [k+2]]-Ka*ET
[22]); // * exp(ks*dist [orderi [k-1]][ orderj [k+2]]);
mataux2[4][2]= I(matw2[4][2])* exp( ks* matw2[4][2]+ kd*D3[orderi [k]][ orderj [k+2]]-Ka*ET
[23]); // * exp(ks*dist [orderi [k]][ orderj [k+2]]);
mataux2[4][3]= I(matw2[4][3])* exp( ks* matw2[4][3]+ kd*D3[orderi [k]+1][ orderj [k+2]]-Ka*ET
[24]); // * exp(ks*dist [orderi [k+1]][ orderj [k+2]]); //
mataux2[4][4]= I(matw2[4][4])* exp( ks* matw2[4][4]+ kd*D3[orderi [k]+2][ orderj [k+2]]-Ka*ET
[25]); // * exp(ks*dist [orderi [k+2]][ orderj [k+2]]);
} // else {mataux2[0][0]= mataux2[0][1]=mataux2[0][2]=mataux2[0][3]=mataux2[0][4]=mataux2[1][0]=
mataux2[1][1]=mataux2[1][2]=mataux2[1][3]=mataux2[1][4]=mataux2[2][0]=mataux2[2][1]=mataux2
[2][3]=mataux2[2][4]=mataux2[3][0]=mataux2[3][1]=mataux2[3][2]=mataux2[3][3]=mataux2[3][4]=
mataux2[4][0]=mataux2[4][1]=mataux2[4][2]=mataux2[4][3]=mataux2[4][4]=1; mataux2[2][2]=50; }
/* mataux2[0][0]= exp(kd*D[orderi [k-2]][ orderj [k-2]])* exp(ks*dist [orderi [k-2]][ orderj [k-2]]);
mataux2[0][1]= exp(kd*D[orderi [k-1]][ orderj [k-2]]) * exp(ks*dist [orderi [k-1]][ orderj [k-2]]); //
mataux2[0][2]= exp(kd*D[orderi [k]][ orderj [k-2]]) * exp(ks*dist [orderi [k]][ orderj [k-2]]);
mataux2[0][3]= exp(kd*D[orderi [k+1]][ orderj [k-2]])* exp(ks*dist [orderi [k+1]][ orderj [k-2]]);
mataux2[0][4]= exp(kd*D[orderi [k+2]][ orderj [k-2]])* exp(ks*dist [orderi [k+2]][ orderj [k-2]]);
mataux2[1][0]= exp(kd*D[orderi [k-2]][ orderj [k-1]])* exp(ks*dist [orderi [k-2]][ orderj [k-1]]);
mataux2[1][1]= exp(kd*D[orderi [k-1]][ orderj [k-1]])* exp(ks*dist [orderi [k-1]][ orderj [k-1]]);
mataux2[1][2]= exp(kd*D[orderi [k]][ orderj [k-1]])* exp(ks*dist [orderi [k]][ orderj [k-1]]);
mataux2[1][3]= exp(kd*D[orderi [k+1]][ orderj [k-1]])* exp(ks*dist [orderi [k+1]][ orderj [k-1]]);
mataux2[1][4]= exp(kd*D[orderi [k+2]][ orderj [k-1]])* exp(ks*dist [orderi [k+2]][ orderj [k-1]]);

```

```

mataux2[2][0]= exp(kd*D[orderi[k-2]][orderj[k]]) * exp(ks*dist[orderi[k-2]][orderj[k]]);
mataux2[2][1]= exp(kd*D[orderi[k-1]][orderj[k]]) * exp(ks*dist[orderi[k-1]][orderj[k]]); //
mataux2[2][2]= exp(kd*D[orderi[k]][orderj[k]]) * exp(ks*dist[orderi[k]][orderj[k]]);
mataux2[2][3]= exp(kd*D[orderi[k+1]][orderj[k]]) * exp(ks*dist[orderi[k+1]][orderj[k]]);
mataux2[2][4]= exp(kd*D[orderi[k+2]][orderj[k]]) * exp(ks*dist[orderi[k+2]][orderj[k]]);
mataux2[3][0]= exp(kd*D[orderi[k-2]][orderj[k+1]]) * exp(ks*dist[orderi[k-2]][orderj[k+1]]); //
mataux2[3][1]= exp(kd*D[orderi[k-1]][orderj[k+1]]) * exp(ks*dist[orderi[k-1]][orderj[k+1]]);
mataux2[3][2]= exp(kd*D[orderi[k]][orderj[k+1]]) * exp(ks*dist[orderi[k]][orderj[k+1]]);
mataux2[3][3]= exp(kd*D[orderi[k+1]][orderj[k+1]]) * exp(ks*dist[orderi[k+1]][orderj[k+1]]);
mataux2[3][4]= exp(kd*D[orderi[k+2]][orderj[k+1]]) * exp(ks*dist[orderi[k+2]][orderj[k+1]]); //
mataux2[4][0]= exp(kd*D[orderi[k-2]][orderj[k+2]]) * exp(ks*dist[orderi[k-2]][orderj[k+2]]);
mataux2[4][1]= exp(kd*D[orderi[k-1]][orderj[k+2]]) * exp(ks*dist[orderi[k-1]][orderj[k+2]]);
mataux2[4][2]= exp(kd*D[orderi[k]][orderj[k+2]]) * exp(ks*dist[orderi[k]][orderj[k+2]]);
mataux2[4][3]= exp(kd*D[orderi[k+1]][orderj[k+2]]) * exp(ks*dist[orderi[k+1]][orderj[k+2]]); //
mataux2[4][4]= exp(kd*D[orderi[k+2]][orderj[k+2]]) * exp(ks*dist[orderi[k+2]][orderj[k+2]]); */
}
if (smat == 9) {
    for (t1=0; t1 <=2; t1++)
        for (t2=0; t2 <=2; t2++) somataux += mataux[t1][t2];
    for (t1=0; t1 <=2; t1++)
        for (t2=0; t2 <=2; t2++) mat3[t1][t2]= mataux[t1][t2]/somataux;
    for(t=0; t <9; t++) {
        soma+=mat3[t/3][t%3];
        acc3[t]=soma;
    }
} else {
    for (t1=0; t1 <=4; t1++)
        for (t2=0; t2 <=4; t2++) somataux2 += mataux2[t1][t2];
    for (t1=0; t1 <=4; t1++)
        for (t2=0; t2 <=4; t2++) mat5[t1][t2]= mataux2[t1][t2]/somataux2;
    for(t=0; t <25; t++) {
        soma+=mat5[t/5][t%5];
        acc5[t]=soma;
    }
}
}

void Desenha(void) {
    if (exibesimulacao) {
        glMatrixMode(GL_MODELVIEW);
        glLoadIdentity();
        // Limpa a janela de visualização com a cor de fundo especificada
        glClearColor(GL_COLOR_BUFFER_BIT);
        // Especifica que a cor corrente é vermelha
        //      R      G      B
        glColor3f(0.7f, 0.7f, 0.8f);
        // Desenha n pontos com a cor corrente
    }
}

```

```

g=0;
refaz:
;
japassou1=japassou2=japassou3=1;
temp=0;
for(r1=0; r1 <=200; r1++) {
    for(r2=0; r2 <=200; r2++) {
        D1[r1][r2] = 0;
        D2[r1][r2] = 0;
        D3[r1][r2] = 0;
    }
}
g++;
n2=0; //Numero total de celulas ativas
for(i=0; i <200; i++) front[i]=posfronti[i]=posfrontj[i]=0;

for(i=0; i <nx; i++)
    for(j=0; j <ny; j++) tab[i][j]=0;
n2 = int(nx*ny*pInicial);
for (k=0; k <n2; k++) {
repsorteio:
;
i= int(rnd(idum)*nx);
if (i==nx) i = nx-1;
j = int(rnd(idum)*ny);
if (j==ny) j = ny-1;
if ((tab[i][j]==1)|| (traj[i][j]==0)) goto repsorteio;
tab[i][j]=1;
orderi[k]=i; //Inicializa a ordem de processamento
orderj[k]=j;
}
//for(i=0;i <nx;i++){ //Inicializa os estados das celulas
// for(j=0;j <ny;j++){
//     dist[i][j]=100000; //inicializa as distancias para o loop abaixo
//     if(rnd(idum)<pInicial && (traj[i][j] != 0 )){ // Define se a posição será ocupada ou não.
//         tab[i][j]=1;
//         orderi[n2]=i; //Inicializa a ordem de processamento
//         orderj[n2]=j;
//         n2++;
//     }
//     else tab[i][j]=0;
// }
// }
// }

for(i=0; i <200; i++) for(j=0; j <200; j++) {

```

```

        trajx[i][j] = traj[i][j]; //trajx=traj;
        distt1[i][j]=distt2[i][j]=distt3[i][j]=1000;
    }
for(k=0; k<3; k++) { //cria 3 códigos direcionais
    if (k==1) for(i=0; i<200; i++) for(j=0; j<200; j++) traj[i][j] = traj2[i][j]; //traj=traj2
    else if (k==2) for(i=0; i<200; i++) for(j=0; j<200; j++) traj[i][j] = traj3[i][j];
    q=0;
    for(i=0; i<200; i++) for(j=0; j<200; j++) dist[i][j]=100000;
    for(i=0; i<nx; i++) { //Inicializa os estados das celulas
        for(j=0; j<ny; j++) {
            if (traj[i][j]>90) { //verifica se a posição (i,j) é um limite de fronteira (porta
                ou fim de corredor)
                front[q]= traj[i][j]-90;
                posfronti[q]=i;
                posfrontj[q]=j;
                q++;
            }
        }
    }

    for(i=0; i<nx; i++) //calcula a distancia de cada célula para a fronteira;
        for(j=0; j<ny; j++) {
            for(tl=0; tl<q; tl++) {
                if (traj[i][j]==front[tl]) {
                    aux3 = sqrt( pow(i-posfronti[tl],2) + pow(j-posfrontj[tl],2) ); //
                        distancia euclidiana até a fronteira do sub ambiente
                    if (dist[i][j]> aux3) dist[i][j]=aux3;
                }
                if (front[tl]==saidaf-90)
                    if (k==0) {
                        aux3 = sqrt( pow(i-posfronti[tl],2) + pow(j-posfrontj[tl],2) );
                            //distancia euclidiana até a saída definitiva (
                                ambiente 1)
                        if (distt1[i][j]> aux3) distt1[i][j]=aux3;
                    } else if (k==1) {
                        aux3 = sqrt( pow(i-posfronti[tl],2) + pow(j-posfrontj[tl],2) );
                            //distancia euclidiana até a saída definitiva (
                                ambiente 2)
                        if (distt2[i][j]> aux3) distt2[i][j]=aux3;
                    } else {
                        aux3 = sqrt( pow(i-posfronti[tl],2) + pow(j-posfrontj[tl],2) );
                            //distancia euclidiana até a saída definitiva (
                                ambiente 2)
                        if (distt3[i][j]> aux3) distt3[i][j]=aux3;
                    }
            }
            if (traj[i][j]>90) dist[i][j]=0.0001;
        }
    }
}

```

```

for(i=0; i<nx; i++) // calcula o numero de passos até a saída definitiva
  for(j=0; j<ny; j++) {
    if ((traj[i][j] > 0)) {
      if (k==0) distc1[i][j] = dist[i][j];
      else if (k==1) distc2[i][j]=dist[i][j];
      else distc3[i][j]=dist[i][j];
      auxi = i;
      auxj = j;
      while ((traj[auxi][auxj] != saidaf-90)&&(traj[auxi][auxj] != saidaf)) {
        aux3 = 100000000;
        for(tl=0; tl<q; tl++) {
          if (traj[auxi][auxj]>90) auxtraj = traj[auxi][auxj]-90;
          else auxtraj = traj[auxi][auxj];
          if (auxtraj==front[tl])
            for (mi = -1; mi<2; mi++)
              for (mj = -1; mj<2; mj++)
                if ((auxtraj != traj[posfronti[tl]+
                  mi][posfrontj[tl]+mj])&&(traj[
                  posfronti[tl]+mi][posfrontj[tl]
                  +mj]<90)&&(traj[posfronti[tl]
                  +mi][posfrontj[tl]+mj] > 0))
                  {
                    if (dist[posfronti[tl]+mi][
                      posfrontj[tl]+mj]<
                      aux3) aux3 = dist[
                        posfronti[tl]+mi][
                          posfrontj[tl]+mj];
                    auxii = posfronti[tl]+mi;
                    auxjj = posfrontj[tl]+mj;
                  }
                }
          if (k==0) distc1[i][j] = distc1[i][j] + aux3;
          else if (k==1) distc2[i][j] = distc2[i][j] + aux3;
          else distc3[i][j] = distc3[i][j] + aux3;
          auxi = auxii;
          auxj = auxjj;
        }
      }
    }
  }

for(i=0; i<nx; i++) //Inicializa os estados das celulas
  for(j=0; j<ny; j++)
    //if ((dist[i-1][j-1]==dist[i-1][j]) || (dist[i-1][j]==dist[i-1][j+1]) || (dist[i-1][j]
    -1]==dist[i-1][j+1]) || (dist[i-1][j-1]==dist[i][j-1]) || (dist[i][j-1]==dist[i
    +1][j-1]) || (dist[i-1][j-1]==dist[i+1][j-1]) || (dist[i+1][j-1]==dist[i+1][j]) || (
    dist[i+1][j]==dist[i+1][j+1]) || (dist[i][j+1]==dist[i+1][j+1])) // verifica
    se há repetições de distancia. Se sim analisará apenas as vizinhanças
    horizontais e verticais. se não, analisara as posições nordeste, sudeste,

```

```

noroeste , sudoeste.
if ((dist[i-1][j-1]==dist[i-1][j]) || (dist[i-1][j]==dist[i-1][j+1]) || (dist[i-1][j
+1]==dist[i][j+1]) || (dist[i][j+1]==dist[i+1][j+1]) || (dist[i+1][j+1]==dist[i
+1][j]) || (dist[i+1][j]==dist[i+1][j-1]) || (dist[i+1][j-1]==dist[i][j-1]) || (dist
[i][j-1]==dist[i-1][j-1])){

mp[0]=dist[i][j-1];
mp[1]=dist[i+1][j];
mp[2]=dist[i][j+1]; // guarda as distancias num vetor para pegar apenas o
menor e o maior
mp[3]=dist[i-1][j];
menord = 100000.0;
maior = 0;
for(r3=0; r3 < 4; r3++) {
    if ((mp[r3] < menord) && (mp[r3]!=0.0001)) menord = mp[r3];
    if (dist[i][j]==1)
        if (mp[r3] < menord) menord = mp[r3]; // pega a menor
        distancia horizontal e vertical
}
for(r3=0; r3 < 4; r3++) if ((mp[r3] > maior) && (mp[r3]!=100000)) maior =
mp[r3]; // pega a maior distancia
// if (traj[i][j] < 90)
if ((traj[i][j] < 90)&&(dist[i][j]!=1))
    if (menord == dist[i-1][j]) codigo[i][j]=4;
    else if (menord == dist[i][j-1]) codigo[i][j]=2; // se estiver
    dentro do domínio, direciona a matriz de probabilidades para a
    posição de menor distancia da fronteira.
    else if (menord == dist[i][j+1]) codigo[i][j]=3;
    else if (menord == dist[i+1][j]) codigo[i][j]=1;

if ((traj[i][j] < 90)&&(dist[i][j]==1))
    if ((menord == dist[i-1][j])&& ((traj[i-1][j]-90) == traj[i][j]) )
        codigo[i][j]=4;
    else if ((menord == dist[i][j-1])&& ((traj[i][j]-90) == traj[i][j
]) ) codigo[i][j]=2; // se estiver dentro do dominio,
    direciona a matriz de probabilidades para a posição de menor
    distancia da fronteira.
    else if ((menord == dist[i][j+1])&& ((traj[i][j+1]-90) == traj[i][j
]) ) codigo[i][j]=3;
    else if ((menord == dist[i+1][j])&& ((traj[i+1][j]-90) == traj[i][j
])) codigo[i][j]=1;

if (traj[i][j] > 90)
    if ((maior==dist[i-1][j]) && (traj[i-1][j]!=traj[i][j]-90)) codigo[
i][j]=4;
    else if ((maior == dist[i][j-1]) && (traj[i][j]-90)!=traj[i][j]-90))
        codigo[i][j]=2;
    else if ((maior == dist[i][j+1]) && (traj[i][j+1]!=traj[i][j]-90))
        codigo[i][j]=3; // se estiver na fronteira, direciona a

```

```

        matriz de probabilidades para a posição de maior distancia da
        fronteira , fazendo a célula sair do dominio.
        else if ((maior == dist[i+1][j]) && (traj[i+1][j]!=traj[i][j]-90))
            codigo[i][j]=1;
    } else {
        mp[0]=dist[i-1][j-1];
        mp[1]=dist[i+1][j-1]; // pega a menor distancia a noroeste , nordeste ,
        // sudeste , sudoeste. Direciona a matriz de probabilidades para a posição
        // de menor distancia da fronteira .
        mp[2]=dist[i-1][j+1];
        mp[3]=dist[i+1][j+1];
        menord = 1000000.0;
        for(r3=0; r3 < 4; r3++) if ((mp[r3] < menord) && (mp[r3]!=0.0001)) menord =
            mp[r3]; //

        if (menord == dist[i+1][j-1]) codigo[i][j]=5;
        else if (menord == dist[i-1][j+1]) codigo[i][j]=8;
        else if (menord == dist[i+1][j+1]) codigo[i][j]=6;
        else if (menord == dist[i-1][j-1]) codigo[i][j]=7;
    }

if (k==0) for(i=0; i<200; i++) for(j=0; j<200; j++) codigo1[i][j] = codigo[i][j]; //codigo1=codigo;
else if (k==1) for(i=0; i<200; i++) for(j=0; j<200; j++) codigo2[i][j] = codigo[i][j]; //codigo2=
    codigo;
else for(i=0; i<200; i++) for(j=0; j<200; j++) codigo3[i][j] = codigo[i][j];
}

for(i=0; i<200; i++) for(j=0; j<200; j++) traj[i][j] = trajx[i][j]; //traj=trajx;

for(j=0; j<ny; j++) {
    for(i=0; i<nx; i++)
        fprintf(sait , " %d" ,codigo3[i][j]);
    fprintf(sait , "\n");
}

fclose(sait);
for(k=0; k<n2; k++) distant[k]=100000;
contret=0;
for(k=0; k<n2; k++) {
    total = 1/distc1[orderi[k]][orderj[k]] + 1/diste2[orderi[k]][orderj[k]] + 1/diste3[orderi[k]][orderj
    [k]];
    // //////////////////////////////////////
    /*     escolha = rnd(idum);
    if (escolha < 1/distc1[orderi[k]][orderj[k]]/total) sentpop[k]=1;
    else if (escolha < 1/distc1[orderi[k]][orderj[k]]/total + 1/diste2[orderi[k]][orderj[k]]/total
    ) sentpop[k]=2;
    else sentpop[k]=3;*/

    // //////////////////////////////////////
    if ((1/distc1[orderi[k]][orderj[k]]> 1/diste2[orderi[k]][orderj[k]])&&(1/diste1[orderi[k]][orderj[k]
    ]> 1/diste3[orderi[k]][orderj[k]])) {

```

```

        sp = (1/distc1[orderi[k]][orderj[k]])/total;
        sc1 = 1/distc2[orderi[k]][orderj[k]];
        sc2 = 1/distc3[orderi[k]][orderj[k]];
        sentpr = 1;
        sentnovo1 = 2;
        sentnovo2 = 3;
    } else if ((1/distc2[orderi[k]][orderj[k]]> 1/distc1[orderi[k]][orderj[k]])&&(1/distc2[orderi[k]][
        orderj[k]]> 1/distc3[orderi[k]][orderj[k]])) {
        sp = (1/distc2[orderi[k]][orderj[k]])/total;
        sc1 = 1/distc1[orderi[k]][orderj[k]];
        sc2 = 1/distc3[orderi[k]][orderj[k]];
        sentpr = 2;
        sentnovo1 = 1;
        sentnovo2 = 3;

    } else {
        sp = (1/distc3[orderi[k]][orderj[k]])/total;
        sc1 = 1/distc1[orderi[k]][orderj[k]];
        sc2 = 1/distc2[orderi[k]][orderj[k]];
        sentpr = 3;
        sentnovo1 = 1;
        sentnovo2 = 2;

    }

    pp1 = pow(sp,kr2);
    pp2 = (1-pp1)*(sc1/(sc1+sc2));
    // pp3 = (1-pp1)*sc2/(sc1+sc2);
    escolha = rnd(idum);
    if (escolha<pp1) sentpop[k]=sentpr;
    else {
        if (escolha < pp1+pp2) sentpop[k]=sentnovo1;
        else sentpop[k]=sentnovo2;
    }
}

////////////////////////////////////
peoples = n2;
for(i=0; i<n2; i++) {
    situacao[i]=0;
    id[i]=0;
}

for(i=1; i<=ng; i++) { //define o centro do grupo
    volta;;
    sort[i] = (int) (rnd(idum)*n2);
    if (situacao[sort[i]]==1) goto volta;
    situacao[sort[i]]= 1;
    id[sort[i]]= i;
}

```



```

for(k=0; k<n2; k++) // define quais serão as células de grupo
    if (situacao[k]==1){
        contel=1;
        for(k2=0; k2<n2; k2++)
            if ((traj[orderi[k]][orderj[k]]==traj[orderi[k2]][orderj[k2]])&&(abs(orderi[k]-
                orderi[k2])<=15)&&(abs(orderj[k]-orderj[k2])<=15)&&(contel<4)&&(situacao[k2
                ]==0) ) {
                    contel++;
                    situacao[k2]=2;
                    id[k2]=id[k];
                    sentpop[k2]=sentpop[k];
            }
        }
//     situacao[k2]=2; // célula dependente;
//     celobj[k2]=id[k]; // célula que seguirá (objetivo);
//     sentpop[k2]=sentpop[k];
//     ndep[k]++;
//     }

////////////////////////////////////

nsims=0;
iniciodes:
;
nsims++;
for(r1=1; r1<=200; r1++) {
    for(r2=1; r2<=200; r2++) {
        DeltaD = D1[r1-1][r2-1]+D1[r1-1][r2]+D1[r1+1][r2]+D1[r1][r2-1]+D1[r1][r2+1]+D1[r1+1][r2+1]+
            D1[r1+1][r2-1]+D1[r1-1][r2+1]-8*D1[r1][r2];
        D1[r1][r2] = (1-gamma)*D1[r1][r2]+(alpha*((1-gamma)/8))*DeltaD;
        // if (D1[r1][r2]>500) D1[r1][r2]=500;
        DeltaD = D2[r1-1][r2-1]+D2[r1-1][r2]+D2[r1+1][r2]+D2[r1][r2-1]+D2[r1][r2+1]+D2[r1+1][r2+1]+
            D2[r1+1][r2-1]+D2[r1-1][r2+1]-8*D2[r1][r2];
        D2[r1][r2] = (1-gamma)*D2[r1][r2]+(alpha*((1-gamma)/8))*DeltaD;;
        // if (D2[r1][r2]>500) D2[r1][r2]=500;
        DeltaD = D3[r1-1][r2-1]+D3[r1-1][r2]+D3[r1+1][r2]+D3[r1][r2-1]+D3[r1][r2+1]+D3[r1+1][r2+1]+
            D3[r1+1][r2-1]+D3[r1-1][r2+1]-8*D3[r1][r2];
        D3[r1][r2] = (1-gamma)*D3[r1][r2]+(alpha*((1-gamma)/8))*DeltaD;
    }
}

for(k=0; k<n2; k++) { //embaralha a ordem de processamento
    auxi=orderi[k];
    auxj=orderj[k];
    sentaux = sentpop[k];
    situaux = situacao[k];
}

```

```

codigaux = id[k];
trocak=(int)(rnd(idum)*n2);
orderi[k]=orderi[trocak];
orderj[k]=orderj[trocak];
orderi[trocak]=auxi;
orderj[trocak]=auxj;
sentpop[k]=sentpop[trocak];
situacao[k] = situacao[trocak];
id[k] = id[trocak];
sentpop[trocak]= sentaux;
situacao[trocak] = situaux;
id[trocak] = codigaux;
}

for (i=0; i<=200; i++)
    for (j=0; j<=200; j++)
        for (k=0; k<=9; k++)
            CA[i][j][k]=0;

for(k=0; k<n2; k++) {
    auxi=orderi[k];
    auxj=orderj[k];
    conter=0;
    if (sentpop[k]==1) //população normal
        faux = codigo1[orderi[k]][orderj[k]];
    else if (sentpop[k]==2) faux = codigo2[orderi[k]][orderj[k]];
    else faux = codigo3[orderi[k]][orderj[k]];
    switch (faux) {
        case 1:
            for (j=(auxj-pe); j<=(auxj+pe); j++)
                for (i=(auxi+1); i<=(auxi+pe); i++)
                    if ((tab[i][j]==1)|| (traj[i][j]==0)) conter++;
            break;
        case 2:
            for (i=(auxi-pe); i<=(auxi+pe); i++)
                for (j=(auxj-1); j>=(auxj-pe); j--)
                    if ((tab[i][j]==1)|| (traj[i][j]==0)) conter++;
            break;
        case 3:
            for (i=(auxi-pe); i<=(auxi+pe); i++)
                for (j=(auxj+1); j<=(auxj+pe); j++)
                    if ((tab[i][j]==1)|| (traj[i][j]==0)) conter++;
            break;
        case 4:
            for (j=(auxj-pe); j<=(auxj+pe); j++)
                for (i=(auxi-1); i>=(auxi-pe); i--)
                    if ((tab[i][j]==1)|| (traj[i][j]==0)) conter++;
            break;
        case 5:

```

```

cont3=0;
for (j=(auxj-pe); j<=(auxj+pe-1); j++) {
    for (i=(auxi-pe+1+cont3); i<=(auxi+pe); i++)
        if ((tab[i][j]==1)|| (traj[i][j]==0)) conter++;
    cont3++;
}
break;
case 6:
cont3=0;
for (j=(auxj+pe); j>=(auxj-pe+1); j--) {
    for (i=(auxi-pe+1+cont3); i<=(auxi+pe); i++)
        if ((tab[i][j]==1)|| (traj[i][j]==0)) conter++;
    cont3++;
}
break;
case 7:
cont3=0;
for (j=(auxj-pe); j<=(auxj+pe-1); j++) {
    for (i=(auxi-pe); i<=(auxi+pe-1-cont3); i++)
        if ((tab[i][j]==1)|| (traj[i][j]==0)) conter++;
    cont3++;
}
break;
case 8:
cont3=0;
for (j=(auxj+pe); j>=(auxj-pe+1); j--) {
    for (i=(auxi-pe); i<=(auxi+pe-1-cont3); i++)
        if ((tab[i][j]==1)|| (traj[i][j]==0)) conter++;
    cont3++;
}
break;
}
cont=0;
cont2=0;
if (situacao[k]>0) sentpopant = sentpop[k];
for (j=(auxj-pe); j<=(auxj+pe); j++)
    for (i=(auxi-pe); i<=(auxi+pe); i++) {
        cont2++;
        if ((tab[i][j]==1)|| (traj[i][j]==0)) conter++;
    }
if (conter == pe+2*pow(pe,2)) if (cont < cont2-re12) {
// if (sentpop[k]==1)
// auxd = distc1[orderi[k]][orderj[k]]/(distc1[orderi[k]][orderj[k]] + distc2[orderi[k]][orderj[k]]);
// else auxd = distc2[orderi[k]][orderj[k]]/(distc1[orderi[k]][orderj[k]] + distc2[orderi[k]][orderj[k]]);
// if (auxd < 0.5) auxd = pow(auxd,kr); else auxd = 1 - pow(1-auxd,kr);
// if (rnd(idum)<auxd) sentpop[k]=3-sentpop[k];

total = 1/distc1[orderi[k]][orderj[k]] + 1/distc2[orderi[k]][orderj[k]] + 1/distc3[
orderi[k]][orderj[k]];

```

```

if (sentpop[k]==1) {
    sp = (1/distc1[orderi[k]][orderj[k]])/total;
    sc1 = 1/distc2[orderi[k]][orderj[k]];
    sc2 = 1/distc3[orderi[k]][orderj[k]];
    sentnovo1 = 2;
    sentnovo2 = 3;
} else if (sentpop[k]==2) {
    sp = (1/distc2[orderi[k]][orderj[k]])/total;
    sc1 = 1/distc1[orderi[k]][orderj[k]];
    sc2 = 1/distc3[orderi[k]][orderj[k]];
    sentnovo1 = 1;
    sentnovo2 = 3;

} else {
    sp = (1/distc3[orderi[k]][orderj[k]])/total;
    sc1 = 1/distc1[orderi[k]][orderj[k]];
    sc2 = 1/distc2[orderi[k]][orderj[k]];
    sentnovo1 = 1;
    sentnovo2 = 2;
}

pp1 = pow(sp,kr);
pp2 = (1-pp1)*sc1/(sc1+sc2);
// pp3 = (1-pp1)*sc2/(sc1+sc2);
escolha = rnd(idum);
if ((pp1 <= escolha) && (escolha < pp1+pp2))
    sentpop[k]=sentnovo1;
else if ( pp1+pp2 <= escolha) sentpop[k]=sentnovo2;
} else {
    conta=0;
    contb=0;
    contc=0;
    for (j=(auxj-pe); j<=(auxj+pe); j++)
        for (i=(auxi-pe); i<=(auxi+pe); i++) {
            if (traj[i][j]==0) {
                conta++;
                contb++;
                contc++;
            }
            if ((i!=0)&&(j!=0))
                for(m=0; m<n2; m++)
                    if ((orderi[m]==i)&&(orderj[m]==j))
                        if (sentpop[m]==1) conta++;
                        else if (sentpop[m]==2) contb++;
                        else contc++;
        }
    if ((sentpop[k]==1)&&(contb+contc) >=cont2-rel))
        if (rnd(idum)<probsm) {
            if (contb > contc) sentpop[k]=2;

```

```

        else sentpop[k]=3;
    }
    if ((sentpop[k]==2)&&( (conta+conte) >=cont2-rel))
        if (rnd(idum)<probsm) {
            if (conta > conte) sentpop[k]=1;
            else sentpop[k]=3;
        }
    if ((sentpop[k]==3)&&( (conta+contb) >=cont2-rel))
        if (rnd(idum)<probsm) {
            if (conta > contb) sentpop[k]=1;
            else sentpop[k]=2;
        }
    }

if ((situacao[k]>0)&&(sentpopant != sentpop[k])) {
    ds2=10000;
    for(k2=0; k2<n2; k2++)
        if (id[k]==id[k2]) { // pertence ao mesmo grupo
            if (sentpopant == 1) {
                ds1 = distc1[orderi[k]][orderj[k]];
                if (distc1[orderi[k2]][orderj[k2]]<ds2) ds2= distc1[orderi[k2]][
                    orderj[k2]];
            } else if (sentpopant == 2) {
                ds1 = distc2[orderi[k]][orderj[k]];
                if (distc2[orderi[k2]][orderj[k2]]<ds2) ds2= distc2[orderi[k2]][
                    orderj[k2]];
            } else {
                ds1 = distc3[orderi[k]][orderj[k]];
                if (distc3[orderi[k2]][orderj[k2]]<ds2) ds2= distc3[orderi[k2]][
                    orderj[k2]];
            }
        }
    if (ds1 > ds2) // indica que este não é o mais adiantado do grupo. não pode mudar a direção
        sentpop[k]=sentpopant;
    else
        for(k2=0; k2<n2; k2++) if (id[k]==id[k2]) sentpop[k2]=sentpop[k];
}

for(k=0; k<n2; k++) { //Calcula o campo de antecipação
    if (sentpop[k]==1) faux = codigo1[orderi[k]][orderj[k]];
    else if (sentpop[k]==2) faux = codigo2[orderi[k]][orderj[k]];
    else faux = codigo3[orderi[k]][orderj[k]];
    switch (faux) {
        case 1:
            CA[orderi[k]+1][orderj[k]][1]++; //soma 1 ponto para a vizinhança 1 passo na
                direção preferencial da célula;
            CA[orderi[k]+2][orderj[k]][1]++;

```

```

        // CA[orderi[k]][orderj[k]+3][1]++;
        break;
    case 2:
        CA[orderi[k]][orderj[k]-1][2]++; //soma 1 ponto para a vizinhança 1 passo na
            direção preferencial da célula;
        if ((orderj[k]-2) >=0) CA[orderi[k]][orderj[k]-2][2]++;
        // if ((orderi[k]-3) >=0) CA[orderi[k]-3][orderj[k]][2]++;
        break;
    case 3:
        CA[orderi[k]][orderj[k]+1][4]++; //soma 1 ponto para a vizinhança 1 passo na
            direção preferencial da célula;
        CA[orderi[k]][orderj[k]+2][4]++;
        // CA[orderi[k]+3][orderj[k]][4]++;
        break;
    case 4:
        CA[orderi[k]-1][orderj[k]][3]++; //soma 1 ponto para a vizinhança 1 passo na
            direção preferencial da célula;
        if ((orderi[k]-2) >=0) CA[orderi[k]-2][orderj[k]][3]++;
        // if ((orderj[k]-3) >=0) CA[orderi[k]][orderj[k]-3][3]++;
        break;
    case 5:
        CA[orderi[k]+1][orderj[k]-1][5]++; //soma 1 ponto para a vizinhança 1 passo na
            direção preferencial da célula;
        if ((orderi[k]-2) >=0) CA[orderi[k]+2][orderj[k]-2][5]++;
        // if ((orderi[k]-3) >=0) CA[orderi[k]-3][orderj[k]+3][5]++;
        break;
    case 6:
        CA[orderi[k]+1][orderj[k]+1][6]++; //soma 1 ponto para a vizinhança 1 passo na
            direção preferencial da célula;
        CA[orderi[k]+2][orderj[k]+2][6]++;
        // CA[orderi[k]+3][orderj[k]+3][6]++;
        break;
    case 7:
        CA[orderi[k]-1][orderj[k]-1][7]++; //soma 1 ponto para a vizinhança 1 passo na
            direção preferencial da célula;
        if ((orderi[k]-2) >=0) if ((orderj[k]-2) >=0) CA[orderi[k]-2][orderj[k]-2][7]++;
        // if ((orderi[k]-3) >=0) if ((orderj[k]-3) >=0) CA[orderi[k]-3][orderj[k]-3][7]++;
        break;
    case 8:
        CA[orderi[k]-1][orderj[k]+1][8]++; //soma 1 ponto para a vizinhança 1 passo na
            direção preferencial da célula;
        if ((orderj[k]-2) >=0) CA[orderi[k]-2][orderj[k]+2][8]++;
        // if ((orderj[k]-3) >=0) CA[orderi[k]+3][orderj[k]-3][8]++;
        break;
    }
}

for(k=0; k<n2; k++) { //varredura
iniciovarr:

```

```

:
i00=orderi[k];
j00=orderj[k];
if (sentpop[k]==1) auxk = traj[i00][j00];
else if (sentpop[k]==2) auxk = traj2[i00][j00];
else auxk = traj3[i00][j00];
if(auxk == saidaf) { // out
    if(rnd(idum)<pSai) {
        tab[i00][j00]=0;
        n2--;
        if(n2>k) {
            orderi[k]=orderi[n2];
            orderj[k]=orderj[n2];
            sentpop[k]=sentpop[n2];
            situacao[k] = situacao[n2];
            id[k] = id[n2];
            goto iniciovarr;
        }
    }
}

else {
    volta = 'N';
    if (sentpop[k]==1) { //população normal;
        if (D1[orderiant[k]][orderjant[k]] > 0) {
            volta='S'; //retira o efeito do próprio rastro
            D1[orderiant[k]][orderjant[k]] -=1;
        }
        redefine(codigo1[orderi[k]][orderj[k]],1);
        if (volta == 'S') D1[orderiant[k]][orderjant[k]] +=1;
    } else if (sentpop[k]==2) {
        if (D2[orderiant[k]][orderjant[k]] > 0) {
            volta='S'; //retira o efeito do próprio rastro
            D2[orderiant[k]][orderjant[k]] -=1;
        }
        redefine(codigo2[orderi[k]][orderj[k]],2);
        if (volta == 'S') D2[orderiant[k]][orderjant[k]] +=1;
    } else {
        if (D3[orderiant[k]][orderjant[k]] > 0) {
            volta='S'; //retira o efeito do próprio rastro
            D3[orderiant[k]][orderjant[k]] -=1;
        }
        redefine(codigo3[orderi[k]][orderj[k]],2); //ver o dois se é mesmo
        if (volta == 'S') D2[orderiant[k]][orderjant[k]] +=1;
    }
    orderiantaux = orderi[k];
    orderjantaux = orderj[k];

    //calulca a nova metrica: numero de vezes que a célula parou ou voltou

```

```

if (sentpop[k]==1) {
    if (distant[k]<= distc1[orderi[k]][orderj[k]]) contret++;
    distant[k] = distc1[orderi[k]][orderj[k]];
} else if (sentpop[k]==2) {
    if (distant[k]<= distc2[orderi[k]][orderj[k]]) contret++;
    distant[k] = distc2[orderi[k]][orderj[k]];
} else {
    if (distant[k]<= distc3[orderi[k]][orderj[k]]) contret++;
    distant[k] = distc3[orderi[k]][orderj[k]];
}
mov(2, i00, j00, &ni, &nj);
//////////
if (situacao[k]>0) { //Verifica se é celula de grupo e so se movimenta se não afastar do
    grupo
    marc=0; //maredis=0;
    imin=jmin=201;
    imax=jmax=-1;
    ds2=-1;
    for(k2=0; k2<n2; k2++)
        if (id[k2]==id[k]) { //pertence ao mesmo grupo
            if (sentpop[k]==1) {
                //aux1 = traj[orderi[k]][orderj[k]];
                //aux2 = traj[orderi[k2]][orderj[k2]];
                ds1 = distc1[orderi[k]][orderj[k]];
                if (distc1[orderi[k2]][orderj[k2]]>ds2) ds2= distc1[orderi[
                    k2]][orderj[k2]];
            } else if (sentpop[k]==2) {
                //aux1 = traj2[orderi[k]][orderj[k]];
                //aux2 = traj2[orderi[k2]][orderj[k2]];
                ds1 = distc2[orderi[k]][orderj[k]];
                if (distc2[orderi[k2]][orderj[k2]]>ds2) ds2= distc2[orderi[
                    k2]][orderj[k2]];
            } else {
                //aux1 = traj3[orderi[k]][orderj[k]];
                //aux2 = traj3[orderi[k2]][orderj[k2]];
                ds1 = distc3[orderi[k]][orderj[k]];
                if (distc3[orderi[k2]][orderj[k2]]>ds2) ds2= distc3[orderi[
                    k2]][orderj[k2]];
            }
            // if (aux1 > 90) aux1 = aux1-90;
            // if (aux2 > 90) aux2 = aux2-90;
            // if (aux1 < aux2) marc=1;
            if (orderi[k2] < imin) imin = orderi[k2];
            if (orderj[k2] < jmin) jmin = orderj[k2];
            if (orderi[k2] > imax) imax = orderi[k2];
            if (orderj[k2] > jmax) jmax = orderj[k2];
        }
    if (ni<=imin) imin=ni;
}

```



```

if (nj<=jmin) jmin=nj;
if (ni>=imax) imax=ni;
if (nj>=jmax) jmax=nj;

// if ((imax-imin,2)+pow(jmax-jmin,2)> pow(16,2))&&(marc==0)://{ ni=orderiantaux; nj=
orderjantaux;}
if (abs(imax-imin+1)*abs(jmax-jmin+1)>16){ //16 representa o eta: area do CG se for
verdade, houve um distanciamento das células do grupo
//for(k2=0; k2<n2; k2++)
if (ds2 == ds1) //é o elemento mais afastado do grupo
{
// if ((rnd(idum)>0.999)&&!(traj[orderi[k]][orderj[k]]==traj2[orderi[k]][
orderj[k]]==traj3[orderi[k]][orderj[k]])) { //0,99 indica a prob
do elemento não pertencer mais ao grupo por estar distante. Elimina-se
então a espera
if (rnd(idum)>0.999){
id[k]=0;
situacao[k]=0;
}
}
else if ((ds1+4)<ds2) { // Se for verdade, a célula está muito adiantada e precisa
parar
tab[ni][nj]=0;
tab[orderiantaux][orderjantaux]=1;
ni=orderiantaux;
nj=orderjantaux;
}
}
}
// if (marc==2) {redefine(1,4); mov(2, i00, j00, &ni, &nj);}

orderi[k]=ni;
orderj[k]=nj;

if((orderiantaux != orderi[k]) || (orderjantaux != orderj[k])) {
orderiant[k] = orderiantaux;
orderjant[k] = orderjantaux;
if (sentpop[k]==1) D1[orderiant[k]][orderjant[k]]+= 1;
else if (sentpop[k]==1) D2[orderiant[k]][orderjant[k]]+= 1;
else D3[orderiant[k]][orderjant[k]]+= 1;
}

/* else {
for(k2=0;k2<n2;k2++)
if (celobj[k]==id[k2]){ //procura a celula objetivo
if ((orderi[k2] < orderi[k])&&(orderj[k2]<orderj[k])) redefine(7,3);// primeiro quadrante
else if ((orderi[k2] < orderi[k])&&(orderj[k2] = orderj[k])) redefine(2,3);//mesma
coluna linha anterior
}
}
}

```

```

else if ((orderi[k2] < orderi[k])&&(orderj[k2] > orderj[k])) redefine(5,3)//
segundo quadrante
else if ((orderi[k2] = orderi[k])&&(orderj[k2] < orderj[k])) redefine(4,3); //
mesma linha coluna anterior
else if ((orderi[k2] > orderi[k])&&(orderj[k2] < orderj[k])) redefine
(8,3); //terceiro quadrante
else if ((orderi[k2] > orderi[k])&&(orderj[k2] = orderj[k]))
redefine(3,3);//mesma coluna linha posterior
else if ((orderi[k2] > orderi[k])&&(orderj[k2] > orderj[k]))
redefine(6,3);//quarto quadrante
else if ((orderi[k2] = orderi[k])&&(orderj[k2] > orderj[k
])) redefine(1,3);//mesma linha coluna posterior

mov(2, i00, j00, &ni, &nj);
orderi[k]=ni;
orderj[k]=nj;
}
}*/
}
i00 = orderi[k];
j00 = orderj[k];
if (exibesimulacao) {
    if (situacao[k] >0) plote(i00,j00,pixsize,7);
    else if (sentpop[k] == 1) plote(i00,j00,pixsize,2);
    else if (sentpop[k] == 2)plote(i00,j00,pixsize,9);
    else plote(i00,j00,pixsize,13);
} //else plote(i00,j00,pixsize,11);
}
/****** */
if (exibesimulacao) {
    // for(i=0;i<d1*1000000000;i++)
    // for(j=0;j<d1*1000000000;j++){
// Sleep(d1*30);

// for(i=0;i<ny;i++){
// for(j=0;j<nx;j++){
// plote(i,j,pixsize,tab[i][j]);

gIFlush();
for(i=0; i<nx; i++)
    for(j=0; j<ny; j++) {
        aux = traj[i][j];
        if (aux != 0) aux =1;
        else aux = 4;
        plote(i,j,pixsize,aux);
    }
}
}
}

```

```

/* ***** */
if (japassou1)
    if (n2<=(peoples*0.75)) {
        fprintf(sai, " %d", nsims);
        japassou1=0;
    }
if (japassou2)
    if (n2<=(peoples*0.5)) {
        fprintf(sai, " %d", nsims);
        japassou2=0;
    }
if (japassou3)
    if (n2<=(peoples*0.25)) {
        fprintf(sai, " %d", nsims);
        japassou3=0;
    }

//
// fprintf(sai, "%d ", n2);
if ((n2>0)&&(nsims<10000)) goto iniciodes; //if ((n2>0)&&(nsims<5030)) goto iniciodes;
temp = (nsims*0.298142)/60;
fprintf(sai, " %d %f %d %d %d\n", g, temp, nsims, peoples, contret); // tempo total de evacuação, numero de
    simulações e numero de pessoas no dominio
somatemp += temp;
if (g<replicacoes) goto refaz;
media = somatemp/replicacoes;
fprintf(sai, "tempo medio de evacuação total: %f minutos\n", media);

fprintf(sai, " ");
/* for(i=0;i<=256;i++){
    for(j=0;j<=256;j++){
        fprintf(sai, " %d ".D[j][i]);
    }
    fprintf(sai, "\n");
} */
fim:
;

}
// Inicializa parâmetros de rendering
void Inicializa (void) {
    // Define a cor de fundo da janela de visualização como preta
    glClearColor(0.0f, 0.0f, 0.0f, 0.0f);
    //glClearColor(100, 20, 180, 1.0f);
}

// Função callback chamada quando o tamanho da janela é alterado
void AlteraTamanhoJanela(GLsizei w, GLsizei h) {

```

```

// Evita a divisao por zero
if(h == 0) h = 1;
// Especifica as dimensões da Viewport
glViewport(50, 100, w, h);
// Inicializa o sistema de coordenadas
glMatrixMode (GL_PROJECTION);
glLoadIdentity ();
// Estabelece a janela de seleção (left, right, bottom, top)
if (w <= h)
    gluOrtho2D (4.0f, 200.0f, 0.0f, 200.0f*h/w);
else
    gluOrtho2D (4.0f, 200.0f*w/h, 0.0f, 200.0f);
}
// Programa Principal
int main(int argc, char *argv[]) {
    // printf("concluido");
    somatemp=0;
    par=fopen("parametros.txt", "r");
    // fscanf(par,"%s", layout);
    fscanf(par, "%s", trajetoria); // trajetoria principal
    fscanf(par, "%s", trajetoria2); // trajetoria secundária
    fscanf(par, "%s", trajetoria3);
    // fscanf(par,"%s", campo);
    fscanf(par, "%d",&nx);
    fscanf(par, "%d",&ny);
    fscanf(par, "%d",&pixsize);
    fscanf(par, "%d",&depth);
    fscanf(par, "%d",&nsimsmax);
    fscanf(par, "%d",&lancemax);
    fscanf(par, "%d",&lanceenmax);
    fscanf(par, "%d",&pause1);
    fscanf(par, "%d",&pause2);
    fscanf(par, "%d",&printflag);
    fscanf(par, "%d",&ntries);
    fscanf(par, "%d",&idum);
    fscanf(par, "%lg",&pEntra);
    fscanf(par, "%lg",&pSai);
    fscanf(par, "%lg",&pInicial);
    fscanf(par, "%d",&smat);
    fscanf(par, "%d",&replicacoes);
    fscanf(par, "%d",&exibesimulacao);
    fscanf(par, "%d",&saidaf);
    fclose(par);

    par=fopen("parametros2.txt", "r");
    fscanf(par, "%d",&d1);
    fscanf(par, "%lg",&kr);
    fscanf(par, "%lg",&Ka);
    fscanf(par, "%lg",&kd);
}

```

```

fscanf(par, "%lg",&ks);
fscanf(par, "%lg",&probsm);
fscanf(par, "%d",&rel1);
fscanf(par, "%d",&rel2);
fscanf(par, "%lg",&gamma);
fscanf(par, "%lg",&alppa);
fscanf(par, "%d",&pe);
fscanf(par, "%d",&ng);
fclose(par);

leiadir=fopen(trajetoria, "r");
for(i=0; i<ny; i++)
    for(j=0; j<nx; j++)
        fscanf(leiadir, "%d",&traj[j][i]);
fclose(leiadir);
leiadir=fopen(trajetoria2, "r");
for(i=0; i<ny; i++)
    for(j=0; j<nx; j++)
        fscanf(leiadir, "%d",&traj2[j][i]);
fclose(leiadir);
leiadir=fopen(trajetoria3, "r");
for(i=0; i<ny; i++)
    for(j=0; j<nx; j++)
        fscanf(leiadir, "%d",&traj3[j][i]);
fclose(leiadir);
// leiacampo=fopen(campo, "r");
// for(i=0; i<ny; i++)
//     for(j=0; j<nx; j++)
//         fscanf(leiacampo, "%lg",&S[i][j]);
// fclose(leiacampo);
// for(j=0; j<ny; j++)
//     for(i=0; i<nx; i++)
//         ocup[i][j]=1;

//inicialização aqui (coisas que só são acionadas uma única vez)
sai=fopen("saida.txt", "w");
sait=fopen("saidat.txt", "w");
if (exibesimulacao) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(1200,700);
    glutInitWindowPosition(10,10);
    glutCreateWindow("Simulação de tráfego de pessoas ");
    glutDisplayFunc(Desenha);
    glutReshapeFunc(AlteraTamanhoJanela);
    Inicializa();
    glutMainLoop();
} else Desenha();
fclose(sai);

```

```

        printf("concluido");
    }
    double logpow(double x1, double x2) { /* power(2,3)=8 */
        return( (log(x1)*x2) );
    }
    double power(double x1, double x2) { /* power(2,3)=8 */
        return( exp(log(x1)*x2) );
    }
    double sqr( double x ) {
        return (x*x);
    }
    int moverer(int j) {
        double auxr;
        auxr=rnd(idum);
        if(auxr<0.25) return(j-1);
        else if(auxr<0.75) return(j);
        else return(j+1);
    }
    int movehor(int i) {
        double auxr;
        auxr=rnd(idum);
        if(auxr<0.10) return(i-1);
        else if(auxr<0.20) return(i);
        else return(i+1);
    }
    int mov(int range, int i0, int j0, int *ni, int *nj) {
        double aux1,aux2;
        int p,q,i,j,k,cont=0,i00,j00,i0min,i0max,j0min,j0max;
        i0min=i0-2*range;
        if(i0min<0)i0min=0;
        i0max=i0+2*range;
        if(i0max>=nx)i0max=nx-1;
        j0min=j0-2*range;
        if(j0min<0)j0min=0;
        j0max=j0+2*range;
        if(j0max>=ny)j0max=ny-1;
        for(i=i0min; i<=i0max; i++) // Inicializa mark[][]
            for(j=j0min; j<=j0max; j++)
                mark[i][j]=0;
        i0min=i0-range;
        if(i0min<0)i0min=0;
        i0max=i0+range;
        if(i0max>=nx)i0max=nx-1;
        j0min=j0-range;
        if(j0min<0)j0min=0;
        j0max=j0+range;
        if(j0max>=ny)j0max=ny-1;
        for(i=i0min; i<=i0max; i++) { //Localiza as ativas proximas
            for(j=j0min; j<=j0max; j++) {

```

```

        if(tab[i][j]==1 && i!=i0 && j!=j0)
            for(lance=0; lance<lancemax; lance++) {
                i00=i00+i;
                j00=j00+j;
                for(lancelen=0; lancelen<lancelenmax; lancelen++) {
                    if(traj[i00][j00] == 0) { //fronteira do dominio
                        fbound(i00,j00,&i00,&j00);
                        mark[i00][j00]++;
                    } else { //dentro do dominio
                        finner(i00,j00,&i00,&j00);
                        mark[i00][j00]++;
                    }
                }
            }
    }
}

if(printflag) {
    fprintf(sai, "%d %d:\n", i0, j0);
    for(i=i0min; i<=i0max; i++) {
        for(j=j0min; j<=j0max; j++) fprintf(sai, "%d ", mark[i][j]);
        fprintf(sai, "\n");
    }
    fprintf(sai, "\n");
}

r=rnd(idum);
k=0;
if(smat==9) {
    while(k<smat && r>acc3[k])k++;
    *ni=k/3-1+ i0;
    *nj=k%3-1+ j0;
    if(*ni!=i0 || *nj!=j0) {
        tab[i0][j0]=0;
        if(*ni<0 || *ni>=nx || *nj<0 || *nj>=ny || mark[*ni][*nj]>0 ||
            tab[*ni][*nj]>0 || (traj[*ni][*nj])==0) {
            *ni=i0;
            *nj=j0;
            tab[i0][j0]=1;
        } else {
            tab[*ni][*nj]=1;
        }
    }
} else { //smat==25
    while(k<smat && r>acc5[k])k++;
    *ni=k/5-2+ i0;
    *nj=k%5-2+ j0;
    if(*ni!=i0 || *nj!=j0) {
        tab[i0][j0]=0;
        if(*ni<0 || *ni>=nx || *nj<0 || *nj>=ny || mark[*ni][*nj]>0 ||
            tab[*ni][*nj]>0 || (traj[*ni][*nj])==0) {

```

```

        *ni=i0;
        *nj=j0;
        tab[i0][j0]=1;
    } else {
        tab[*ni][*nj]=1;
    }
}
}
}
void finner(int i000, int j000, int *i00, int *j00) {
    int i,j,k;
    double r;
    r=rnd(idum);
    k=0;
    if(smat==9) {
        while(k<smat && r>acc3[k])k++;
        *i00=k/3-1+ *i00;
        *j00=k%3-1+ *j00;
    } else { //smat==25
        while(k<smat && r>acc5[k])k++;
        *i00=k/5-2+ *i00;
        *j00=k%5-2+ *j00;
    }
}
void fbound(int i000, int j000, int *i00, int *j00) {
    int i,j,k,tent=0,tentmax=5;
    double r;
    do {
        r=rnd(idum);
        k=0;
        if(smat==9) {
            while(k<smat && r>acc3[k])k++;
            i=k/3-1+ *i00;
            j=k%3-1+ *j00;
        } else { //smat==25
            while(k<smat && r>acc5[k])k++;
            i=k/5-2+ *i00;
            j=k%5-2+ *j00;
        }
        tent++;
    } while((tab[i][j] & 1)!=1 && tent<tentmax);
    if(tent<tentmax) {
        *i00=i;
        *j00=j;
    }
}
float rnd(long idum0) {
    long k;
    float ans;

```



```

do {
    idum0 ^= MASK;
    k=(idum0)/IQ;
    idum0=IA*(idum0-k*IQ)-IR*k;
    if(idum0<0) idum0+=IM;
    ans=AM*(idum0);
    idum0^=MASK;
    idum=idum0;
} while(ans>0.999999);
return(ans);
}

void plote(int x, int y, int t, int c) {
    int i, j, a, b;
    double R=0.0, G=0.0, B=0.0;
    if(c==8) glColor3f(1.0, 1.0, 1.0);
    else {
        if(c>0) {
            R=G=B=0.3;
        }
        c/=2;
        if(c%2)R+=0.7;
        c/=2;
        if(c%2)G+=0.7;
        c/=2;
        if(c%2)B+=0.7;
        glColor3f(R, G, B);
    }
    a=x*t;
    b=y*t;
    glBegin(GL_QUADS);
    glVertex2i(a+10, b+10);
    glVertex2i(a+t+10, b+10);
    glVertex2i(a+t+10, b+t+10);
    glVertex2i(a+10, b+t+10);
    glEnd();
}

```