



SENSORIAMENTO ADAPTATIVO DE EVENTOS EM REDES DE AGENTES AUTÔNOMOS MÓVEIS

Rodrigo Roma Esch

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientadores: Valmir Carneiro Barbosa
Fábio Protti

Rio de Janeiro
Dezembro de 2012

SENSORIAMENTO ADAPTATIVO DE EVENTOS EM REDES DE AGENTES
AUTÔNOMOS MÓVEIS

Rodrigo Roma Esch

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO
ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE
ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE
JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A
OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE
SISTEMAS E COMPUTAÇÃO.

Examinada por:

Prof. Valmir Carneiro Barbosa, Ph.D.

Prof. Fábio Protti, D.Sc.

Prof. José Ferreira de Rezende, Ph.D.

Prof. Débora Christina Muchaluat Saade, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
DEZEMBRO DE 2012

Esch, Rodrigo Roma

Sensoriamento adaptativo de eventos em redes de agentes autônomos móveis/Rodrigo Roma Esch. – Rio de Janeiro: UFRJ/COPPE, 2012.

XI, 68 p.: il.; 29, 7cm.

Orientadores: Valmir Carneiro Barbosa

Fábio Protti

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2012.

Referências Bibliográficas: p. 67 – 68.

1. Sensores móveis autônomos. 2. Maximização distribuída. 3. Rede de sensores. 4. Detecção de eventos dinâmicos. I. Carneiro Barbosa, Valmir *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Aos pais maravilhosos que tenho.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

SENSORIAMENTO ADAPTATIVO DE EVENTOS EM REDES DE AGENTES AUTÔNOMOS MÓVEIS

Rodrigo Roma Esch

Dezembro/2012

Orientadores: Valmir Carneiro Barbosa

Fábio Protti

Programa: Engenharia de Sistemas e Computação

Em uma região onde ocorrem eventos de acordo com uma certa densidade ϕ , há diversas aplicações nas quais é desejável que o maior número de eventos possível seja detectado. A densidade ϕ é dinâmica, ou seja, além de variar com a posição, a densidade também varia com o tempo. Para sensoriar a região, há disponíveis N agentes autônomos móveis idênticos. Para maximizar o número de eventos detectados para qualquer cenário de eventos possível, o algoritmo proposto nesta dissertação mantém alguns agentes sensoriando os locais com maior densidade de eventos enquanto que alguns agentes permanecem andando aleatoriamente pelo resto da região a procura de novos locais com incidência de eventos.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

ADAPTATIVE EVENT SENSING IN NETWORKS OF AUTONOMOUS
MOBILE AGENTS

Rodrigo Roma Esch

December/2012

Advisors: Valmir Carneiro Barbosa
Fábio Protti

Department: Systems Engineering and Computer Science

In a region where events occur according to a certain density ϕ , there are many applications in which it's interesting that the largest possible number of events be detected. The density ϕ is dynamic, that is, besides varying according to position, it also varies according to time. To sense the region, there are N identical autonomous mobile agents available. To maximize the number of detected events in any possible scenario, the algorithm proposed in this dissertation keeps some agents sensing the places with largest event density while some agents move randomly through the rest of the region searching for new places with incidence of events.

Sumário

Lista de Figuras	x
Lista de Tabelas	xi
1 Introdução	1
1.1 Motivação	1
1.2 Proposta	2
1.3 Organização desta dissertação	3
2 Trabalhos Relacionados	4
2.1 Information-driven distributed coverage algorithms for mobile sensor networks	4
2.2 Performance-guaranteed distributed coverage control for robotic visual sensor network with limited energy storage	5
2.3 UREA: Uncovered Region Exploration Algorithm for Reorganization of Mobile Sensor Nodes to Maximize Coverage	5
2.4 Mobile Sensor Network Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem	6
2.5 Distributed Cooperative Coverage Control of Sensor Networks	6
2.6 Distributed Coverage Control and Data Collection With Mobile Sensor Networks	7
3 Definindo o Problema	8
3.1 Os eventos	8
3.2 Objetivo	10
3.3 Como mensurar os resultados	10
3.3.1 Parcela de eventos sensorizados por pelo menos um agente	11
3.3.2 Parcela média de eventos que os agentes tomam conhecimento	12
4 Solução Proposta	14
4.1 Os agentes	14
4.2 Ideia geral do algoritmo	15

4.2.1	O caso mais simples possível	16
4.2.2	Um exemplo um pouco mais complexo	17
4.2.3	Economizando energia	18
4.2.4	O modo aleatório e o modo gradiente	19
4.2.5	As mensagens trocadas pelos agentes	20
4.2.6	Tamanho dos passos	22
4.2.7	Quando o agente troca seu modo de execução	23
4.2.8	O terceiro modo de execução	31
4.2.9	Calculando a densidade dinâmica de eventos	32
4.3	Parâmetros	34
4.3.1	Largura e altura da região	34
4.3.2	Número de subdivisões da largura e altura	34
4.3.3	Tempo simulado	34
4.3.4	Tempo de sensoriamento	35
4.3.5	Número mínimo de passos no modo aleatório	35
4.3.6	Gradiente mínimo de F pra sair do modo aleatório	35
4.3.7	Gradiente mínimo de F necessário para permanecer no modo gradiente	36
4.3.8	Aleatoriedade para sair do modo gradiente	36
4.3.9	Gradiente máximo de F para sair do modo gradiente	36
4.3.10	Módulo mínimo de \vec{v}	37
4.3.11	Número de passos no modo de repulsão	37
4.3.12	Número de agentes	37
4.3.13	Tamanho do passo aleatório	38
4.3.14	Tamanho do passo de referência no modo gradiente	38
4.3.15	Raio de sensoriamento	39
4.3.16	Raio de comunicação	39
4.3.17	Probabilidade de recebimento de mensagem	39
4.3.18	Posição inicial	39
4.3.19	Tempo de visibilidade	40
4.3.20	Tempo de interesse	40
4.4	A função F	40
4.5	Algoritmo	44
5	Experimentos	47
5.1	Chuva	48
5.1.1	Parâmetros	48
5.1.2	Resultados	50
5.1.3	Resultados na internet	52

5.1.4	Comentários	52
5.2	Combate à poluição	52
5.2.1	Parâmetros	52
5.2.2	Resultados	55
5.2.3	Resultados na internet	56
5.2.4	Comentários	56
5.3	Chuva de granizo	57
5.3.1	Parâmetros	57
5.3.2	Resultados	59
5.3.3	Resultados na internet	60
5.3.4	Comentários	60
6	Conclusão	62
6.1	Contribuições	62
6.2	Limitações e trabalhos futuros	64
	Referências Bibliográficas	67

Lista de Figuras

4.1	A densidade de eventos mais simples possível.	16
4.2	Agentes distribuídos em um cenário de eventos mais simples possível.	16
4.3	Densidade de eventos dinâmica.	17
4.4	Possível distribuição dos agentes.	18
4.5	Movimento dos agentes.	22
4.6	Densidade de eventos para ilustrar um dos motivo pelos quais é desejável que os agentes não permaneçam no modo gradiente infinitamente.	25
4.7	Exemplo do que acontece quando um agente de centro executa o movimento Browniano.	27
4.8	Possíveis passos do agente ao escolher o modo aleatório.	28
4.9	Exemplo do cálculo de \vec{v} . O vetor resultante está fora de escala.	29
4.10	Exemplo do cálculo de \vec{v} para um agente no centro.	30
4.11	Densidade de eventos dinâmica.	33
5.1	Densidade de eventos idealizada. Os eventos reais utilizados estão disponíveis na página do experimento.	49
5.2	Evolução do número de eventos detectados com o tempo.	51
5.3	Evolução da parcela média de eventos que os agentes conhecem.	51
5.4	Densidade de eventos idealizada. Os eventos reais utilizados estão disponíveis na página do experimento.	53
5.5	Identificadores.	53
5.6	Evolução do número de eventos detectados com o tempo	55
5.7	Evolução da parcela média de eventos que os agentes tomam conhecimento.	56
5.8	Densidade de eventos idealizada. Os eventos reais utilizados estão disponíveis na página do experimento.	57
5.9	Evolução do número de eventos detectados com o tempo.	59
5.10	Evolução da parcela média de eventos que os agentes tomam conhecimento.	60

Lista de Tabelas

5.1	Chuva: Tabela de parâmetros.	50
5.2	Chuva: Eventos sensorizados por pelo menos um agente.	50
5.3	Chuva: Parcela média de eventos que os agentes tomam conhecimento.	51
5.4	Poluição: Tabela de parâmetros.	54
5.5	Poluição: Eventos sensorizados por pelo menos um agente.	55
5.6	Poluição: Parcela média de eventos que os agentes tomam conhecimento.	55
5.7	Granizo: Tabela de parâmetros.	58
5.8	Granizo: Eventos sensorizados por pelo menos um agente.	59
5.9	Granizo: Parcela média de eventos que os agentes tomam conhecimento.	59

Capítulo 1

Introdução

1.1 Motivação

Com os avanços tecnológicos e o desenvolvimento de tecnologias que tornam a comunicação, o processamento e o sensoriamento cada vez mais baratos, o interesse pelo sensoriamento realizado através de agentes autônomos móveis tem crescido nos últimos anos [1].

Há diversas aplicações onde é importante realizar o sensoriamento de determinada região. Como exemplo, podemos citar operações de busca e recuperação, manipulações em ambientes hostis, vigilância e monitoramento do nível de poluição ambiental [2].

Em muitos casos, não é possível monitorar uma região com apenas um sensor devido à complexidade de tal tarefa: imagine uma região muito grande de forma que construir um único sensor capaz de monitorar eventos em toda sua extensão seja algo extremamente custoso ou até mesmo inviável. Imagine também que, caso tal sensor possa ser construído, haveria um único ponto de falha capaz de comprometer todo o projeto.

Para contornar essa situação, a tarefa de monitorar uma grande região pode ser distribuída para diversos agentes autônomos móveis capazes de se comunicar e com capacidades de sensoriamento individual relativamente pequenas mas que, quando seus esforços são somados, o resultado é um enorme poder de sensoriamento capaz de detectar a maioria dos eventos que acontecem na região.

Por serem móveis, os agentes podem ter diversos comportamentos: podem se mover aleatoriamente, podem se mover em direção aos locais com maiores incidências de eventos, podem se mover por locais sem eventos a procura de novos eventos que possam começar a surgir ali, etc.

Quanto mais adequado o comportamento dos agentes for em relação ao cenário de eventos, maior será, utilizando os mesmos recursos (agentes), a capacidade de sen-

soriamiento total. O objetivo deste trabalho é propor um algoritmo para maximizar a capacidade conjunta de detecção sem que, para isso, seja necessário aumentar o número de agentes.

1.2 Proposta

Neste trabalho, apresenta-se um novo algoritmo que visa maximizar a detecção de eventos em determinada região utilizando, para isso, agentes autônomos móveis capazes de sensoriar uma certa região circular em seu entorno e capazes, também, de se comunicar entre si.

O objetivo é obter um alto índice de eventos sendo detectados sem que, para isso, seja necessário um número relativamente grande de agentes; caso contrário, bastaria distribuí-los uniformemente na região [3] onde os eventos acontecem e mantê-los parados que já conseguiríamos bons resultados.

Para executar um sensoriamiento eficaz mesmo sem um número muito grande de agentes, estes precisam estar distribuídos na região onde os eventos podem acontecer de forma a maximizar a probabilidade de que, dado que um evento ocorreu em algum ponto qualquer de tal região, ele seja percebido por algum dos agentes.

Entretanto, os eventos não ocorrem uniformemente distribuídos na região, eles podem seguir qualquer tipo de padrão: podem ocorrer só a direita, só no centro de forma circular, no quadrante superior direito e no inferior esquerdo, enfim, qualquer padrão é possível. Mais do que isso, o padrão de ocorrência dos eventos (a densidade de eventos) pode alterar com o passar do tempo: uma chuva começa com certa intensidade em certo local e vai se deslocando e alterando sua intensidade com o tempo. O algoritmo apresentado é poderoso o suficiente a ponto de manter uma grande parcela de eventos sendo sensoriados mesmo que a densidade de eventos seja dinâmica.

Para isso, o algoritmo apresentado neste trabalho fará com que alguns agentes se movimentem seguindo em direção aos locais com maior incidência de eventos enquanto que outros se mantenham andando aleatoriamente à procura de eventos que possam surgir em outros locais.

Em vez de manter alguns agentes movimentando-se aleatoriamente, se todo agente seguisse em direção aos locais com maior incidência de eventos (assim como a maioria das propostas já existentes na literatura onde os agentes seguem o gradiente — como por exemplo [2]), todos eles se movimentariam em direção a máximos locais e por ali ficariam permanentemente tornando inviável a detecção de eventos em outros pontos de máximo e de outros eventos que possam surgir afastados dos atuais pontos de máximo (onde todos os agentes já estariam concentrados).

1.3 Organização desta dissertação

Esta dissertação está organizada em 6 capítulos, sendo este o primeiro, contendo a introdução.

No **segundo capítulo**, serão apresentados alguns trabalhos relacionados a este, detalhando as diferenças que este possui em relação aos demais.

No **terceiro capítulo**, será definido formalmente o problema. Será explicado o que é um evento, qual o objetivo deste trabalho e, em seguida, quais as métricas utilizadas para mensurar os resultados obtidos com a implementação.

No **quarto capítulo**, será explicada detalhadamente a solução proposta nesta dissertação para o problema definido no segundo capítulo. Primeiro será explicado o que é um agente, em seguida, será dissertado o comportamento esperado dos agentes de acordo com o padrão de ocorrência dos eventos, incluindo alguns exemplos. Após isso, serão explicados todos os parâmetros de entrada do algoritmo proposto e, por último, será apresentado formalmente o algoritmo.

No **quinto capítulo**, temos alguns dos experimentos que foram realizados para validar a eficácia do algoritmo em detectar eventos. Serão apresentados experimentos com diversas variações nos parâmetros de entrada para comprovar que a solução proposta é capaz de lidar com cenários bastante diversificados. Em todos os casos, o experimento é confrontado com o cenário de controle, ou seja, um cenário onde todos os parâmetros de configuração da simulação são idênticos exceto os parâmetros necessários para que todos os agentes se movimentem aleatoriamente.

Por último, no **sexto capítulo** há a conclusão desta dissertação, apresentando suas limitações e apontando trabalhos futuros que podem ser derivados deste.

Capítulo 2

Trabalhos Relacionados

A seguir, serão apresentados os trabalhos que possuem alguma relação com a proposta desta dissertação. Cada trabalho relacionado está em uma seção exclusiva.

2.1 Information-driven distributed coverage algorithms for mobile sensor networks

Como na disposição inicial dos agentes pode haver agentes que não estejam próximos a nenhum evento, estes não estariam participando da cobertura de eventos bem como não saberiam em qual direção andar para conseguir melhorar a probabilidade conjunta de detecção. O trabalho apresentado em [2] coordena o movimento dos agentes a partir de uma disposição inicial dos mesmos de forma que, quando os agentes pararem de se mover, é garantido que todos estarão participando do sensoriamento.

Para isso, o algoritmo começa elegendo os agentes que serão os líderes. Um líder é o agente com mais informação, ou seja, o(s) agente(s) capaz(es) de sensoriar mais eventos em sua posição inicial. Como a quantidade de informação sensoriada nunca decresce, um líder nunca o deixará de ser.

Os demais agentes adotarão o comportamento de “seguir o líder” até que eles estejam em uma posição tal que já sejam capazes de sensoriar eventos, momento no qual eles passam a se mover na direção que aumenta a probabilidade conjunta de detecção. Veja que este trabalho relacionado utiliza a abordagem gradiente presente em diversos trabalhos (inclusive nesta dissertação), e também uma abordagem de *leader-following* conforme visto em [4].

É assumido que todo agente é capaz de comunicar-se diretamente com qualquer outro agente, ou seja, que o grafo de comunicação é completo e estático. É assumido, também, que a densidade de eventos é estática e que todos os agentes já tem conhecimento prévio dessa densidade.

Esta dissertação começou inspirada neste trabalho, entretanto, logo foi percebido

que a necessidade de conhecimento prévio da densidade de eventos, o modelo de comunicação, tal que todo agente comunica-se com todos os demais, e a densidade de eventos estática poderiam não ser razoáveis em algumas aplicações práticas.

Nesta dissertação, não há a utilização do algoritmo *leader-following* [4]. No lugar disso, os agentes permanecem andando aleatoriamente através de um movimento Browniano [12, 13] até que encontrem os locais com incidência eventos, de forma a permitir que eles passem a andar na direção que maximiza a probabilidade conjunta de detecção. Também não foi assumido, nesta dissertação, que a densidade de eventos é estática, que há conhecimento prévio da mesma e nem que o grafo de comunicação é completo ou estático.

2.2 Performance-guaranteed distributed coverage control for robotic visual sensor network with limited energy storage

Em [5], o trabalho resumido na Seção 2.1 é aprimorado, sendo incluídos fatores como limite de energia e ângulo de visão.

Os agentes possuem uma certa quantidade de energia E_i disponível e a movimentação (tanto o deslocamento quanto a alteração de seu ângulo) consomem esta energia. É assumido, também, que a performance do sensoriamento (o seu alcance máximo), depende da quantidade de energia que se tem disponível de forma que a melhor performance possível do sensoriamento se dá quando a quantidade de energia E_i disponível que se tem armazenada é o máximo que o agente pode armazenar.

Ao contrário deste trabalho, nesta dissertação é assumido que há energia ilimitada disponível e o agente é capaz de sensoriar igualmente em todas as direções.

2.3 UREA: Uncovered Region Exploration Algorithm for Reorganization of Mobile Sensor Nodes to Maximize Coverage

O trabalho apresentado em [3] contém um algoritmo que maximiza a cobertura da região a partir de uma configuração inicial da posição dos agentes. Entretanto, este trabalho não leva em consideração os locais com maior incidência de eventos e o objetivo é manter um sensoriamento uniforme em toda a região independente da densidade de eventos.

É feita uma comparação com o algoritmo *Virtual Forces Algorithm* [6], que possui o mesmo propósito, e é observado que, com o mesmo número de agentes, o algoritmo

UREA converge com um número menor de iterações.

Ao contrário deste trabalho relacionado, nesta dissertação não há interesse em distribuir os agentes uniformemente pela região e sim em concentrá-los nos locais com maiores densidades de eventos, de forma a maximizar a detecção dos mesmos pelos agentes.

2.4 Mobile Sensor Network Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem

Em [7], é apresentado um trabalho que se utiliza de uma abordagem de campo potencial para, a partir de uma configuração inicial onde todos os sensores estão próximos, espalhá-los o mais uniformemente possível por toda a região que se consegue alcançar a partir dessa posição inicial dos agentes.

Uma abordagem similar é apresentada em [8]; porém, cada sensor deve satisfazer a restrição de possuir k vizinhos, onde k é um parâmetro de entrada.

Novamente, ao contrário deste trabalho relacionado, o objetivo do algoritmo proposto nesta dissertação é concentrar os agentes nos locais com maiores densidades de eventos.

2.5 Distributed Cooperative Coverage Control of Sensor Networks

Em [9], é apresentado um algoritmo que, assim como esta dissertação, também é baseado no gradiente e os agentes também são autônomos. Entretanto há uma estação central que deve receber toda a informação sensoriada pelos agentes, algo que esta dissertação não contempla.

O modelo inclui o gasto com energia de forma que este deve ser considerado ao escolher os movimentos dos agentes. Toda informação sensoriada por todo agente deve chegar até a estação central através de uma dessas formas:

- O agente envia a informação através da rede *wireless* e a estação central a recebe diretamente;
- O agente envia a informação através da rede *wireless* para a estação central utilizando outros agentes como intermediários.

Devido às características da comunicação *wireless*, enviar as informações diretamente à estação central pode ser extremamente custoso se comparado ao envio da in-

formação utilizando outros agentes como intermediários. Quanto menor a distância entre os agentes envolvidos na troca de mensagens, menor é a soma do custo da comunicação. Através de um parâmetro, é definida a importância da economia de energia em relação ao sensoriamento.

É feita uma abordagem unicamente gradiente, ou seja, os agentes sempre se movem com o objetivo de maximizar a detecção de eventos (dada sua importância em relação à economia de energia). Desta forma, caso um agente esteja em uma posição tal que a densidade de eventos em seu entorno seja nula, o vetor gradiente por ele calculado terá módulo zero e, sendo assim, ele permaneceria parado. Esta dissertação propõe um algoritmo que contempla situações onde os agentes estejam em uma posição sem nenhum evento ocorrendo próximo a ele.

Este trabalho relacionado sugere que a densidade de eventos conhecida pelos agentes seja atualizada conforme novas observações forem sendo feitas, entretanto, não modela como que essa atualização seria feita. O trabalho apresentado nesta dissertação faz essa modelagem da atualização da informação local da densidade de acordo com as novas observações.

2.6 Distributed Coverage Control and Data Collection With Mobile Sensor Networks

A proposta em [10] apresenta um modelo probabilístico que contempla o trabalho de detecção de fontes de dados (locais com maior incidência de eventos) através de um modelo que permite erros nessa detecção. Além disso, este trabalho relacionado também possui interesse na coleta dos dados emergentes de suas fontes. O algoritmo apresentado permite que não haja nenhum conhecimento prévio da densidade de eventos.

Em [11] é apresentado outro trabalho baseado em um modelo probabilístico mas que contempla apenas o posicionamento dos sensores em um ambiente com obstáculos para maximizar a probabilidade de detectar os eventos de acordo com uma certa função de densidade.

Capítulo 3

Definindo o Problema

3.1 Os eventos

O segundo capítulo desta dissertação começa definindo precisamente o que é um evento, ou seja, o item cuja probabilidade de ser detectado queremos que seja maximizada.

Um **evento** consiste no objeto de interesse do experimento, no que se quer detectar para que, com as informações sobre a ocorrência dos mesmos, seja possível tomar certas decisões ou simplesmente para realizar um estudo sobre seu padrão de ocorrência.

A seguir são apresentados alguns exemplos do que é considerado um evento no contexto deste trabalho:

- Em um experimento de laboratório onde são gerados raios, deseja-se descobrir a densidade de raios por metro quadrado da região onde o experimento será realizado. Neste caso, a ocorrência de cada raio é considerada como sendo um evento;
- Em um ecossistema, pode ser interessante detectar os locais com maior concentração de poluição para que, com isso, sejam enviados agentes despoluidores. Neste exemplo, o evento seria o aparecimento de uma partícula nociva ao ecossistema;
- Em um campo arqueológico onde está sendo executada uma escavação à procura de ossos de dinossauros, é de interesse dos cientistas envolvidos no projeto descobrir com a maior rapidez possível o aparecimento de um novo osso. Neste caso, o aparecimento de cada osso é considerado um evento.

Agora serão analisados alguns aspectos relevantes em relação aos exemplos acima de forma a detalhar precisamente o que é um evento.

Primeiramente, note que, em todos os 3 exemplos, os eventos podem ocorrer em qualquer ponto da região. Dessa forma, tem-se que todo evento ocorre em um ponto q tal que $q \in \mathbb{R}^2$. Da mesma forma, o tempo no qual o evento ocorre (contado a partir do começo do experimento) também é contínuo.

Nota-se, também, que cada evento ocorre em um ponto (x, y) específico, ou seja, os eventos são pontos no plano.

Repare que a densidade de ocorrência eventos pode ser dinâmica com o tempo. Dessa forma, em determinado ponto da região, pode ser que a densidade em certo instante de tempo seja diferente da densidade no mesmo ponto em um momento posterior.

Os agentes são capazes de distinguir cada evento dos demais. Para isso, um agente observa o local onde o evento ocorreu e o seu *timestamp* (tempo no qual o evento ocorreu contando desde o começo do experimento). Consideramos que não haverá dois eventos ocorrendo simultaneamente no mesmo ponto (x, y) e, assim, pode-se identificar unicamente os eventos.

No primeiro exemplo, os agentes detectam os raios através da descarga elétrica gerada pelo mesmo; dessa forma, o agente detecta o raio no momento exato em que ele acontece ou não será mais capaz de detectá-lo. No segundo exemplo, a partícula nociva ao ecossistema, após surgir, permanecerá no sistema por algum tempo até sumir; vamos supor que sejam algumas horas. Durante essas horas, os agentes serão capazes de detectá-la. No último exemplo, após um osso de dinossauro ser encontrado, ele permanecerá naquele local até o final do experimento, de forma que, após encontrado, sempre será possível detectá-lo.

O **tempo de visibilidade** é a característica que determina por quanto tempo os eventos ainda poderão ser detectados pelos agentes a partir do momento em que eles ocorrem. No primeiro exemplo (raios), o tempo de visibilidade é nulo (são eventos instantâneos). No segundo, o tempo de visibilidade é de algumas horas e, no último, é infinito.

Neste trabalho, tratamos a visibilidade do evento como “visível” ou “não visível”, ou seja, supondo um evento com tempo de visibilidade de 60 minutos, a chance de um evento ser detectado por um agente no momento de sua ocorrência é a mesma dele ser detectado após 60 minutos, porém, após 61 minutos ele nunca mais poderá ser detectado por nenhum agente. Isso pode não ser natural em alguns casos onde o “rastros” deixado pelo evento (através do qual é possível detectá-lo) vai desaparecendo (imagine que a ocorrência de um raio seja detectada pela marca deixada no solo; essa marca vai sumindo com o tempo de forma que a cada momento fica mais difícil detectá-la).

3.2 Objetivo

O objetivo do algoritmo proposto neste trabalho é distribuir os agentes na região de forma que a probabilidade de que um evento ocorrido seja detectado pelos agentes seja maximizada e que tal probabilidade continue maximizada mesmo que a densidade varie com o passar do tempo.

Não é garantido que o algoritmo apresentado nesta dissertação é ótimo, porém, conforme resultados das simulações realizadas e que estão presentes no Capítulo 5, dados valores adequados aos parâmetros, é garantido que haverá um grande ganho na detecção de eventos pelos agentes em relação ao cenário onde os agentes se movimentam unicamente de acordo com o movimento Browniano [12, 13] (onde a cada passo dado pelo agente é escolhida uma direção aleatória) ou se movimentam unicamente seguindo o gradiente da função F (que diz a probabilidade de que um evento aleatório que tenha ocorrido seja detectado por pelo menos um agente) como, por exemplo, [9].

Em relação à abordagem onde os agentes apenas seguem o gradiente, em alguns casos este comportamento será mais eficaz do que o comportamento misto proposto neste trabalho. Entretanto, nossa proposta é mais genérica pois:

- Mais cenários de eventos são contemplados com um resultado satisfatório;
- É possível configurar os parâmetros de entrada do algoritmo proposto de modo que os agentes se comportem de modo similar às abordagens onde seguem o gradiente.

Não há uma estação central que seja capaz de se comunicar com os agentes. A proposta presente neste trabalho é que os agentes sejam totalmente autônomos (contrastando com [14]) provendo, assim, maior tolerância a falhas.

3.3 Como mensurar os resultados

Se não há uma estação central responsável por receber os eventos detectados pelos agentes, quando exatamente pode-se dizer que “um evento foi detectado”? Basta que um único agente tenha tomado conhecimento de sua existência ou todos os agente deverão tomar conhecimento do mesmo?

Mais a frente, será visto que há duas formas de um agente tomar conhecimento da ocorrência de um evento:

- O agente pode sensoriar diretamente o evento;

- O agente pode receber a informação da ocorrência de um evento através da rede de comunicação. Ao sensoriar um evento, o agente propaga essa informação para os demais agentes, porém não é garantido que todos receberão a mensagem.

Nas duas subseções a seguir, serão explicadas as duas métricas utilizadas para medir a eficácia do algoritmo proposto.

3.3.1 Parcela de eventos sensorizados por pelo menos um agente

Seja n o número total de eventos que ocorreram durante uma certa simulação. Seja d o número de eventos de tal simulação tais que **pelo menos um agente** foi capaz de sensoriá-lo.

A **parcela de eventos sensorizados por pelo menos um agente** P_s é calculada da seguinte forma:

$$P_s = \frac{d}{n}.$$

Evolução desta métrica no tempo

A métrica apresentada considera todos os eventos que ocorreram durante toda simulação. Dessa forma, tem-se um único valor para avaliar todo o experimento.

Em contrapartida, é interessante avaliar a evolução desta métrica com o passar do tempo para verificar se o algoritmo está gerando uma melhoria nos resultados conforme os agentes vão adquirindo uma visão mais precisa a respeito da densidade de eventos que, a princípio, é totalmente desconhecida. A ideia geral é que, no começo do experimento, onde ainda há pouca informação sobre os locais com maiores incidências de eventos, P_s seja menor do que no final da simulação, momento no qual os agentes terão (no geral) uma visão mais precisa a respeito dos locais com maiores probabilidades de ocorrer um evento.

Para acompanhar a evolução de P_s , basta discretizar o tempo e, para cada um dos intervalos gerados, calcular P_s de forma a obter diversos valores que representam a métrica atual porém em determinado intervalo de tempo.

Discretização da região

Imagine um determinado experimento onde concluiu-se que 80% dos eventos foram detectados por pelo menos um agente. Isso significa que, para qualquer sub-região que seja considerada em relação à região total, 80% dos eventos que ocorreram nessa sub-região foram sensorizados? Não, pois pode acontecer que, por exemplo,

certa sub-região tenha 98% dos eventos detectados e outra tenha apenas 15%. Essa discrepância é perfeitamente possível, pois os agentes podem estar concentrados em certo local de forma que ali haja um sensoriamento muito eficaz, porém haja outros locais com déficit de sensores. Estes locais terão um sensoriamento menos eficaz.

Para que seja possível um estudo mais detalhado a respeito dos locais com um sensoriamento mais ou menos eficiente, ou seja, dos locais onde mais ou menos eventos foram sensoriados por pelo menos um agente, discretiza-se a região em D_L e D_A partes na largura e altura, respectivamente, e calcula-se o valor de P_s para os eventos que ocorreram em cada uma das $D_L D_A$ discretizações.

3.3.2 Parcela média de eventos que os agentes tomam conhecimento

Dependendo da aplicação onde o algoritmo proposto será utilizado, pode ser importante que, para cada evento que ocorreu, o maior número possível de agentes tome conhecimento de sua ocorrência, em vez de ser suficiente que um único agente tome conhecimento (métrica anterior).

Além da importância relativa ao tipo de aplicação que está sendo executada, tem-se também que, quanto maior o conhecimento global da ocorrência de cada um dos eventos, cada agente terá uma visão mais precisa da atual densidade de eventos. Quanto mais precisa for essa visão individual de cada agente, mais corretas serão as suas decisões que dependem dessa informação (mais a frente será visto que, para se movimentar, um agente utiliza a informação local sobre os eventos).

A **parcela média de eventos que os agentes tomam conhecimento** tem o seguinte significado: dado que um evento qualquer ocorreu, quantos agentes, na média, ficarão sabendo de sua ocorrência? Repare que um agente pode tomar conhecimento de um determinado evento através do sensoriamento direto do mesmo ou por ter recebido essa informação de algum outro agente. Quanto melhor for a comunicação entre os agentes, maior será o valor desta métrica.

No melhor dos casos, todos os agentes terão conhecimento de todos os eventos que ocorreram e, portanto, $P_c = 1$ (onde P_c é a métrica em questão). No pior dos casos, $P_c = 0$ pois nenhum agente detectou nenhum evento sequer. Nos casos intermediários, teremos que um certo agente conheceu 43% dos eventos, outro conheceu 97%, outro, 72%, etc. Neste caso, faz-se a média aritmética da parcela de eventos que cada agente conheceu para calcular P_c .

Sendo n o número total de eventos que ocorreram durante a simulação e sendo k_i o número de eventos de que o agente i tomou conhecimento (entre 0 e n), a parcela de eventos de que o agente i tomou conhecimento é

$$\frac{k_i}{n}.$$

Fazendo a média de todos os N agentes, temos que

$$P_c = \frac{1}{N} \sum_{i=1}^N \frac{k_i}{n}.$$

Ou, equivalentemente,

$$P_c = \frac{1}{Nn} \sum_{i=1}^N k_i.$$

Evolução desta métrica no tempo

Assim como no caso da parcela de eventos sensorizados por pelo menos um agente, também é interessante verificar a evolução da parcela média de eventos que os agentes tomam conhecimento com o passar do tempo para verificar o ganho obtido no sensoriamento conforme os agentes adquirem mais e mais informações.

Demais informações sobre esta discretização temporal são análogas à métrica anterior.

Discretização da região

Pelos mesmos motivos citados na métrica anterior, também é interessante verificar o comportamento da parcela média de eventos que os agentes tomam conhecimento de forma discretizada.

Demais informações sobre esta discretização também são análogas à métrica anterior.

Capítulo 4

Solução Proposta

4.1 Os agentes

Para detectar os eventos que estão ocorrendo, utilizam-se diversos agentes distribuídos pela região onde os eventos ocorrem (a forma com que os agentes se distribuem pela região será vista mais adiante ainda neste capítulo). Esta seção definirá os agentes, descrevendo suas características, possibilidades e limitações.

Um agente consiste em um **sensor móvel** capaz de sensoriar uma região circular em seu entorno, cujo raio de sensoriamento é um parâmetro de entrada do simulador. O sensor, assim como os eventos, é tratado como sendo um ponto no plano. Assim como para os eventos, por uma questão de simplicidade, é assumido que todos os sensores (agentes) possuem as mesmas características.

Sendo R_s o alcance máximo do sensoriamento, temos que a probabilidade de o agente i na posição s_i detectar um evento na posição q é (assumindo a mesma função de decaimento assumida em [2]):

$$p_i(q) = \begin{cases} \left(\frac{d_i - R_s}{R_s}\right)^2 & \text{se } d_i \leq R_s \\ 0 & \text{c.c.} \end{cases}$$

onde $d_i = \|q - s_i\|$.

Esse decaimento conforme a distância até o evento aumentar é natural, visto que, quanto mais longe um evento está do sensor, mais difícil será perceber a presença do mesmo.

Os agentes são capazes de se movimentar, porém, neste trabalho, assumiremos que o tempo que cada agente gasta com a locomoção é desprezível pois, mais a frente, durante a explicação do algoritmo, será visto que cada agente, após se movimentar, permanece um enorme tempo parado executando o sensoriamento daquela região, de forma que o tempo gasto movimentando-se poderá ser desconsiderado.

Os agentes também são capazes de se comunicar. Neste trabalho, o modelo

adotado para a comunicação entre os agentes é o seguinte:

- Toda mensagem enviada por um agente tem por objetivo ser recebida por todos os demais agentes pois, conforme será visto adiante, contém informações de interesse de todos;
- Toda mensagem enviada por um agente é enviada em *broadcast* (consequência da tecnologia *wireless* utilizada);
- Quando um agente recebe uma mensagem, se este agente ainda não a tinha recebido antes e se não foi ele o autor da mensagem, ele novamente a propaga em *broadcast*. Caso contrário, ele a ignora;
- Quando um agente envia uma mensagem (seja ele o autor da mensagem ou não), não há confirmação de quantos agentes a receberam;
- Toda mensagem enviada por um agente i na posição s_i será recebida por todo agente j na posição s_j com a seguinte probabilidade:

$$r_i(s_j) = \begin{cases} p_c & \text{se } d_{ij} \leq R_c \text{ e } i \neq j \\ 0 & \text{c.c.} \end{cases}$$

onde $d_{ij} = \|s_i - s_j\|$. p_c e R_c são parâmetros de entrada do algoritmo.

Dessa forma, o grafo de comunicação consiste em um modelo Booleano, que é amplamente utilizado como modelo para comunicação [12, 15], onde a adição de aresta entre dois vértices ocorrerá com probabilidade p_c caso a distância entre os vértices seja de até R_c unidades, caso contrário, eles não serão capazes de se comunicar (ou seja, não haverá aresta entre eles).

Note que, com este modelo adotado, por não contemplar mensagens de ACK, não é garantido nem ao menos que um único agente receberá a mensagem enviada. Porém, dados valores razoáveis para R_c e p_c , nas componentes conexas onde há diversos caminhos entre quaisquer 2 vértices (agentes), há grandes chances de todos os vértices da componente receberem a mensagem. Quanto maior a densidade de agentes, maior tende a ser o número de caminhos entre quaisquer 2 agentes (vértices) e menores poderiam ser os valores de R_c e p_c e ainda assim a probabilidade de recebimento de mensagens continuaria alto.

4.2 Ideia geral do algoritmo

O objetivo desta seção é apresentar informalmente o algoritmo proposto neste trabalho de forma a facilitar o entendimento, em vez de apenas apresentá-lo através de pseudo-código. A versão formal do algoritmo será apresentada na Seção 4.5.

4.2.1 O caso mais simples possível

Para começar a explicar o algoritmo, vejamos o caso mais simples possível. Seja o seguinte cenário mostrado na Figura 4.1, onde os eixos representam a região e a coloração indica a densidade relativa de eventos. Neste exemplo, tem-se um cenário de eventos com densidade estática, ou seja, o padrão de ocorrência de eventos não se altera com o tempo.

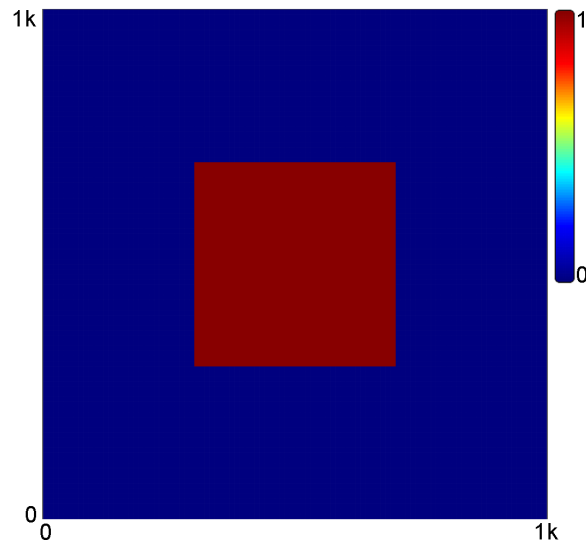


Figura 4.1: A densidade de eventos mais simples possível.

Se sabemos que os eventos sempre ocorrerão unicamente no quadrado no meio da região e se há 16 agentes disponíveis para sensoriar tal região, uma posição dos agentes para que sejam capazes de sensoriar com uma grande eficiência se dá conforme a Figura 4.2 a seguir, onde os agentes estão uniformemente distribuídos:

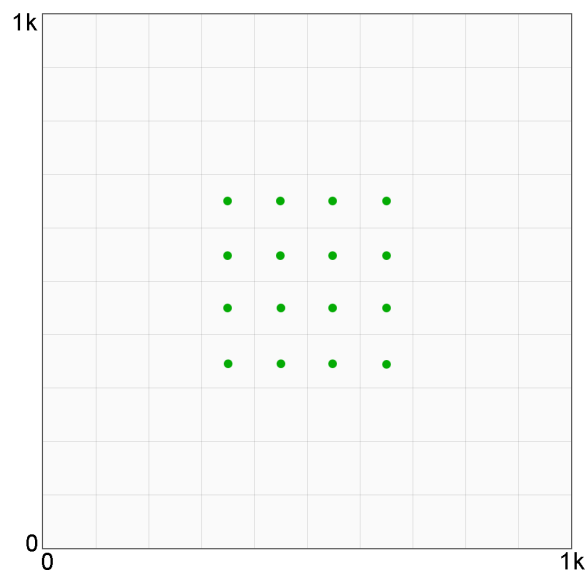


Figura 4.2: Agentes distribuídos em um cenário de eventos mais simples possível.

Neste exemplo, como a densidade de eventos é estática, os agentes podem permanecer nas posições mostradas na Figura 4.2 durante todo o experimento, que ainda assim haverá um grande desempenho no sensoriamento.

4.2.2 Um exemplo um pouco mais complexo

Na Figura 4.3, temos um cenário dinâmico de eventos. O experimento simulará 120 minutos e, nele, nos primeiros 60 minutos, temos que os eventos ocorrem unicamente em uma parte do lado esquerdo da região, conforme mostra a Figura 4.3a. Nos últimos 60 minutos, ocorrem, também, eventos do lado direito com a mesma densidade relativa que o lado esquerdo, conforme mostra a Figura 4.3b.

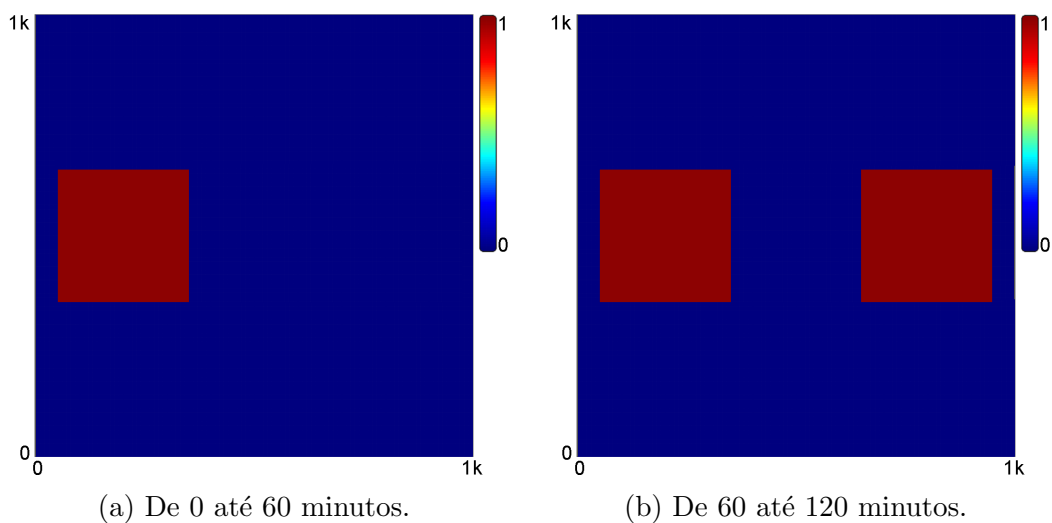


Figura 4.3: Densidade de eventos dinâmica.

Neste caso, os agentes não poderão ser distribuídos estaticamente conforme mostrado no exemplo anterior pois, caso contrário, a sub-região do lado direito que contém eventos após 60 minutos não teria nenhum de seus eventos detectados.

Então qual deve ser o comportamento dos agentes nessa situação? Nossa proposta é que, em situações similares a essa, alguns agentes façam o sensoriamento da sub-região a esquerda (de forma similar ao exemplo anterior), enquanto que alguns outros agentes permanecerão andando aleatoriamente de acordo com o movimento Browniano (a direção de cada passo é escolhida aleatoriamente). Quando esses agentes que estão se movendo aleatoriamente perceberem a presença de novos eventos surgindo do lado direito, eles começarão a sensoriar essa sub-região, assim como os outros agentes que estão sensoriando a sub-região à esquerda.

Dessa forma, após 60 minutos desse exemplo, poderíamos dividir os agentes em três grupos:

- Agentes que estão sensoriando os eventos no lado esquerdo;

- Agentes que estão sensoriando os eventos no lado direito;
- Agentes que ainda estão andando aleatoriamente à procura de eventos em outros locais.

A Figura 4.4 mostra uma possível distribuição dos agentes na situação recém descrita, onde os agentes representados por círculos verdes estão sensoriando uma das duas regiões que contém eventos e os agentes representados por quadrados azuis, andando aleatoriamente à procura de novas sub-regiões contendo eventos.

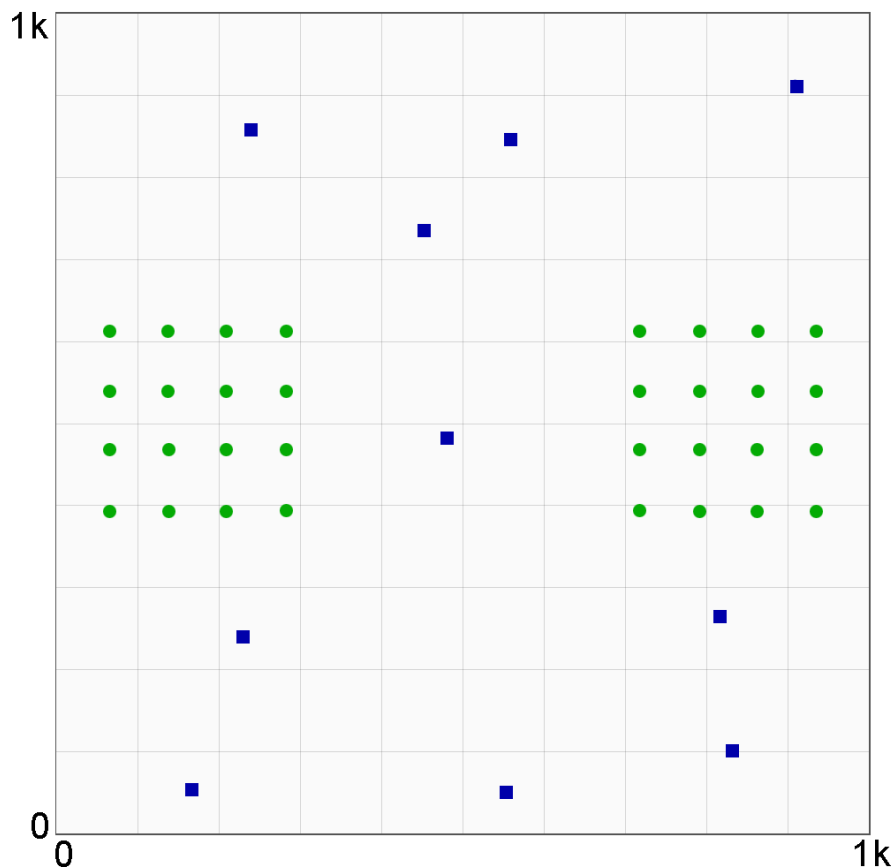


Figura 4.4: Possível distribuição dos agentes.

Repare que, neste exemplo, foram inseridos mais agentes, pois, como os locais onde estão ocorrendo eventos representam uma área maior que no exemplo anterior, bem como ainda são necessários agentes para procurar novas sub-regiões contendo eventos, se utilizássemos o mesmo número de agentes que no exemplo anterior, a performance seria muito inferior.

4.2.3 Economizando energia

Pelo que foi visto até agora, os agentes que estiverem andando aleatoriamente (quadrados azuis no exemplo acima) estarão se movendo 100% do tempo. Mais a frente,

será visto que mesmo os agentes que estiverem sensoriando determinada região onde há eventos ocorrendo (círculos verdes no exemplo acima) também poderão estar se movendo tanto quanto qualquer outro agente.

Entretanto, se um agente permanece em movimento durante todo o tempo, haverá um gasto muito grande de energia com a locomoção. Além do mais, já que cada agente deve informar os eventos que foram por ele sensorizados e também a sua posição aos demais agentes (assim como será visto mais adiante), se, a cada pequena mudança de posição ou a cada evento sensorizado, o agente enviar uma mensagem a ser propagada a todos os demais agentes, haverá, novamente, um grande gasto de energia também para executar a comunicação.

A medida adotada neste trabalho para economizar energia e, com isso, tornar viável a sua implementação em uma gama maior de situações, é fazer com que os agentes não fiquem se movendo durante todo o tempo e somente em uma pequena fração do mesmo.

A fração de tempo em que o agente estará se movimentando será tão pequena em relação ao tempo em que o mesmo estará parado apenas sensoriando a região em seu entorno que tal fração será desconsiderada, ou seja, o movimento será tratado como se fosse instantâneo.

Mais a frente, será visto que, sempre que um agente executa um movimento, o mesmo também envia uma mensagem a ser propagada até atingir toda a componente conexa no grafo de comunicação. Dessa forma, se todos os agentes permanecerem o mesmo tempo parados e, conseqüentemente, se movimentarem ao mesmo tempo de forma síncrona, todos os agentes estariam enviando simultaneamente uma mensagem. Assim, há grande chances de haver colisão de pacotes devido às características do meio *wireless*.

Para diminuir a probabilidade de colisão de pacotes e, com isso, aumentar a eficácia na propagação de mensagens, o tempo que um agente permanecerá parado antes de executar um movimento (e enviar uma mensagem) é uma variável aleatória com distribuição normal. Após executar cada movimento, o agente calcula, de acordo com os parâmetros da distribuição normal especificada na entrada do algoritmo, quanto tempo ele permanecerá parado até executar um novo movimento.

4.2.4 O modo aleatório e o modo gradiente

Na Figura 4.4 mostrada anteriormente, é possível dividir os agentes em duas categorias de acordo com o seu comportamento:

- Agentes que estão andando aleatoriamente (representados por quadrados azuis na Figura 4.4);

- Agentes que estão distribuídos em uma certa sub-região de forma a maximizar a probabilidade de que os eventos que ali estão ocorrendo sejam detectados (representados por círculos verdes na Figura 4.4).

O primeiro comportamento, a partir de agora, será chamado de **modo aleatório**, enquanto que o segundo, de **modo gradiente**. Mais a frente, serão vistas as condições para que os agentes troquem de modo.

No **modo aleatório**, o agente age autonomamente, ou seja, suas decisões não são baseadas em fatores externos a ele tais como a posição dos outros agentes, por exemplo.

Já no **modo gradiente**, além das informações locais, o agente também precisa de algumas informações externas que serão agora explicadas. Lembre-se de que, neste modo, o agente precisa calcular qual a melhor posição para ele ficar de forma a maximizar a probabilidade de detectar eventos. Quando um agente estiver no modo gradiente e for executar um movimento, ele calculará qual a melhor direção e sentido para se movimentar de forma a maximizar o número de eventos sensorizados naquela área.

Seja $s = (s_1, \dots, s_N)$ a posição dos N agentes presentes no experimento. Seja ϕ uma matriz de duas dimensões de dimensões D_L e D_A contendo a densidade relativa de eventos na região discretizada no instante de tempo atual. Seja $F(s, \phi)$ uma função (que não será definida neste momento) que retorna a probabilidade de que, dado que um evento qualquer ocorreu (de acordo com a densidade de eventos ϕ), esse evento seja detectado por algum dos agentes que estão nas posições definidas em s .

De acordo com as definições do parágrafo anterior, quando um agente i no modo gradiente for se movimentar, a direção escolhida será fornecida através da seguinte derivada parcial em relação à posição de i :

$$\frac{\partial F}{\partial s_i}$$

Ou seja, a direção escolhida será o gradiente de F , aquela que maximiza F .

4.2.5 As mensagens trocadas pelos agentes

Conforme visto anteriormente, sempre que um agente se move, ele também envia uma mensagem. A seguir, será explicado o que essas mensagens contêm.

Um agente, quando está no modo aleatório, não precisa de maiores informações. Entretanto, quando o mesmo está no modo gradiente, para se mover, conforme já visto, precisará de duas informações que, pelo que foi visto até agora, ele não possui:

- A localização dos demais agentes, ou seja, o parâmetro $s = (s_1, \dots, s_N)$ da função $F(s, \phi)$;
- A densidade relativa de eventos, ou seja, o parâmetro ϕ de $F(s, \phi)$.

Para suprir a necessidade exposta no primeiro item acima (a posição dos demais agentes), sempre que um agente envia uma mensagem, ele anexa a ela a sua posição. Como sempre que um agente se move ele também envia uma mensagem, em geral, os demais agentes terão a sua posição sempre atualizada, pelo menos os agentes próximos ao agente que se moveu (para calcular $\partial F/\partial s_i$ basta, na verdade, apenas conhecer a posição dos agentes vizinhos ao agente que está calculando tal função).

Caso algum agente se mova, porém algum de seus vizinhos não tenha conseguido atualizar sua posição devido a falha de comunicação, há grandes chances de a informação ser atualizada no próximo passo que o agente der visto que, para que ele dê mais um passo e continue sem atualizar a posição, é necessário que hajam 2 falhas consecutivas e isso se torna mais improvável quanto maior for a probabilidade de sucesso na comunicação.

O segundo item necessário para calcular F é a densidade relativa de eventos ϕ (que é discretizada). A densidade relativa de eventos só pode ser calculada através dos eventos que foram detectados, ou seja, a densidade é inferida verificando os locais com maior e menor incidência de eventos.

Porém, sabe-se que nem todos os eventos serão detectados pelos agentes. Dessa forma, o valor calculado de ϕ não necessariamente representa a realidade, é apenas uma aproximação. Sabe-se, também, que, quanto maior o número de eventos na amostra, maior será a precisão do valor calculado para ϕ .

Se o agente, ao calcular ϕ , utilizar como amostra apenas os eventos detectados por ele próprio, a amostra será muito pequena e limitada aos locais por onde ele percorreu. Isso nos leva a desejar que a amostra seja composta de todos os eventos detectados por qualquer um dos agentes de forma a maximizar a precisão de ϕ .

Conforme visto anteriormente na Seção 4.1, os agentes são capazes de distinguir os eventos através das informações do local e momento em que o mesmo ocorreu.

Concluindo, quando um agente se move, ele também envia uma mensagem com as seguintes informações:

- Sua nova posição;
- A posição e *timestamp* dos eventos que foram detectados desde o seu último movimento.

Lembrando que através da posição e do *timestamp* é possível distinguir um eventos dos demais.

4.2.6 Tamanho dos passos

Foi visto que, de tempos em tempos, o agente executa um movimento e que esse movimento pode ser um passo em qualquer direção (de acordo com o movimento Browniano) ou um passo no sentido do gradiente de F . A seguir, será explicada qual a distância percorrida durante a execução desses passos, lembrando que o tempo gasto se movimentando é desprezível, de forma que, neste trabalho, é assumido que a movimentação é instantânea.

O comportamento dos agentes em relação ao tamanho do passo é diferente quando se está no modo aleatório de quando se está no modo gradiente. No modo aleatório, todo passo tem tamanho fixo de S_a unidades de distância, onde S_a é um parâmetro de entrada do algoritmo. Já no modo gradiente, o tamanho do passo será calculado da seguinte forma, onde S_g é um parâmetro de entrada do algoritmo:

$$S_g \left\| \frac{\partial F}{\partial s_i} \right\|.$$

Ou seja, o passo será proporcional ao módulo do vetor gradiente de F .

Na Figura 4.5, as linhas representam o caminho percorrido pelo agente quando este se moveu da última posição em que esteve até a posição atual.

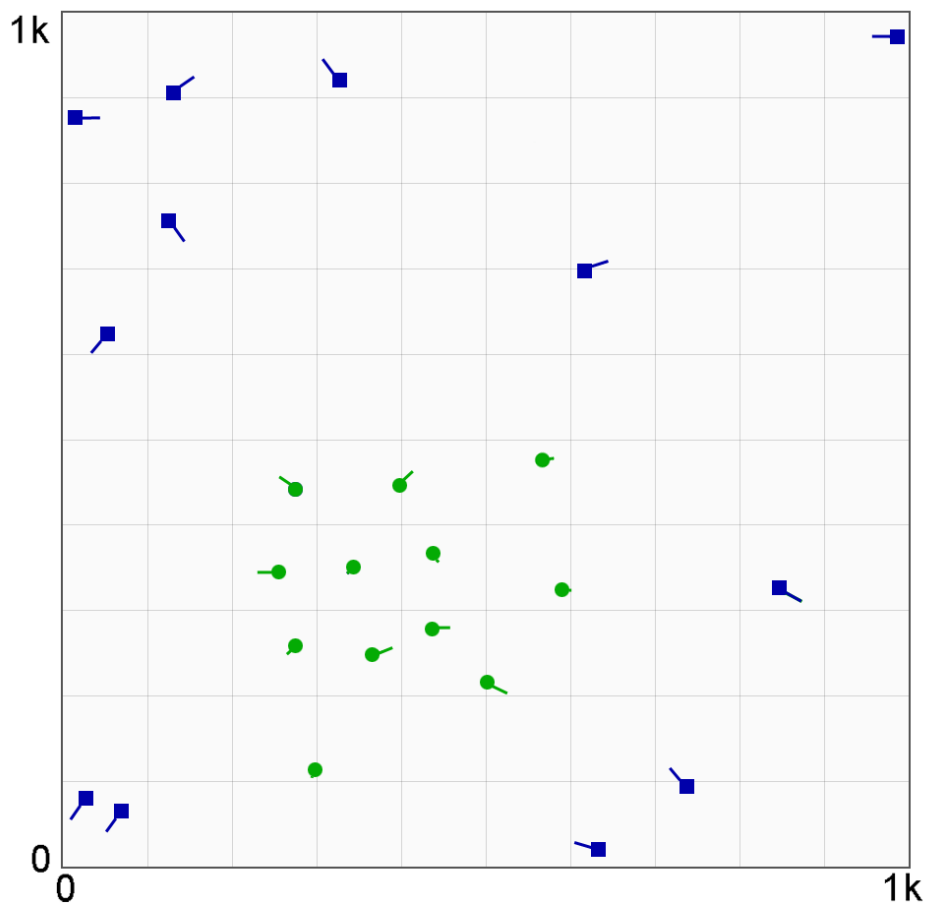


Figura 4.5: Movimento dos agentes.

Repare que os agentes no modo aleatório (os quadrados azuis na figura), possuem todos um rastro de mesmo tamanho. Já os agentes no modo gradiente (círculos verdes) possuem rastros de tamanhos distintos pois tal tamanho depende da intensidade do gradiente de F naquela posição. Alguns rastros são tão pequenos que nem é possível vê-los na figura. Todos os movimentos são feitos em linha reta.

Agentes próximos à fronteira

Um caso particular relacionado à movimentação dos agentes ocorre quando um desses está próximo à fronteira da região, ou seja, está muito próximo à borda que limita a região onde o experimento está ocorrendo, e tal agente deseja movimentar-se.

Nessas situações, a borda da região será tratada como sendo um “espelho” para o movimento do agente, ou seja, o agente poderá escolher movimentar-se para um ponto fora da região. Entretanto, quando o mesmo se deparar com a fronteira durante o seu trajeto, a direção do movimento sendo executado é alterada de forma a “refletir” o movimento.

4.2.7 Quando o agente troca seu modo de execução

Assim que começa o experimento, todos os agentes estão no modo aleatório, visto que não há informações sobre os locais com maior incidência de eventos para que eles saibam onde devem se concentrar.

As perguntas a serem respondidas a seguir são: em que momento um agente troca do modo aleatório para o modo gradiente? Em que momento ele volta ao modo aleatório?

Do modo aleatório para o modo gradiente

Para que um agente pare de andar aleatoriamente e comece a andar seguindo o gradiente de F , há duas condições necessárias. São elas:

- Ter executado pelo menos A_{minS} passos aleatórios consecutivos, ou seja, desde o último instante em que entrou neste modo;
- O módulo do gradiente de F satisfizer $\left\| \frac{\partial F}{\partial s_i} \right\| > A_{threshold}$,

onde A_{minS} e $A_{threshold}$ são parâmetros de entrada do algoritmo e ambas as condições devem ser satisfeitas para haver a mudança de modo. Agora serão analisadas cada uma das condições acima.

A primeira condição (número mínimo de passos) tem por objetivo impedir que um agente que acabou de sair do modo gradiente para o modo aleatório volte a

seguir o gradiente muito rapidamente. Se ele foi para o modo aleatório, é desejável que ele continue explorando novas localidades antes que possa voltar a sensoriar uma determinada localidade específica novamente.

A segunda condição visa garantir que não haverá agentes no modo gradiente em locais onde não estejam ocorrendo eventos, o que não seria desejável.

Seja um agente i em um local sem a incidência de eventos em todo seu entorno. Nessa situação, o agente i não está contribuindo para a probabilidade global de detectar eventos (já que não ocorrem eventos nos locais onde ele é capaz de sensoriar). Se ele não está contribuindo para tal probabilidade, não importa qual seja o deslocamento Δs_i que o agente i venha a fazer, a probabilidade global será a mesma. Ora, se a probabilidade será a mesma, o módulo de $\frac{\partial F}{\partial s_i}$ será nulo.

Para finalizar, temos que $A_{threshold}$ deve ser sempre positivo e que, quanto maior, mais difícil será pra um agente entrar no modo gradiente.

Do modo gradiente para o modo aleatório

Após entrar no modo gradiente, é desejável que o agente possa retornar ao modo aleatório por dois motivos:

- Quando cessar a ocorrência de eventos no local onde o agente está. Caso contrário, o mesmo estaria no modo gradiente em um local onde não estão mais ocorrendo eventos, o que não é desejável. O agente ficaria parado visto que o módulo do gradiente seria nulo;
- Para manter alguns agentes no modo aleatório com o objetivo de explorar outros locais além daqueles onde já há agentes sensoriando no modo gradiente. Para ilustrar melhor, veja a Figura 4.6. Se todos os agentes entrarem no modo gradiente nos primeiros 60 minutos para sensoriar os eventos no lado esquerdo, conforme a Figura 4.6a, após 60 minutos haverá eventos ocorrendo do lado direito (Figura 4.6b) porém nenhum agente será capaz de chegar até lá pois estão “presos” sensoriando o lado esquerdo.

Veja que não é possível que um agente que está no modo gradiente sensoriando os eventos no lado esquerdo da Figura 4.6b chegue até os eventos no lado direito sem sair do modo gradiente, pois, para isso, tem que passar por uma região onde não há nenhum evento ocorrendo e isso faria com que o valor de F diminuísse. Mas como o modo gradiente é caracterizado por fazer os agentes andarem na direção em que o valor de F aumenta, o agente nunca escolheria andar naquela direção.

Dado que são dois os motivos pelos quais é interessante que o agente retorne ao modo aleatório, também são duas as condições suficientes que fazem um agente voltar para tal modo:

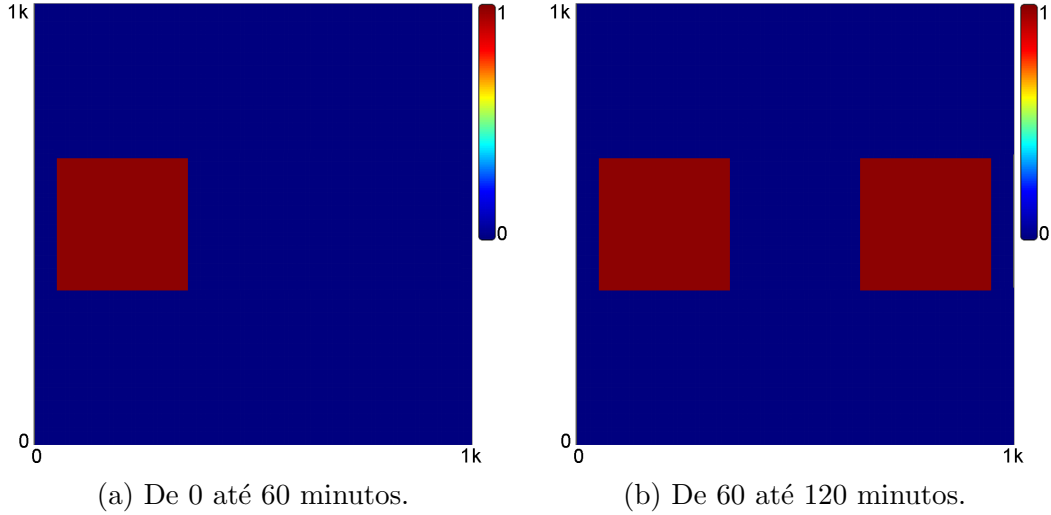


Figura 4.6: Densidade de eventos para ilustrar um dos motivos pelos quais é desejável que os agentes não permaneçam no modo gradiente infinitamente.

- O módulo do gradiente de F satisfizer $\|\partial F/\partial s_i\| < G_{threshold}$
- Uma segunda condição formada pelas três sub-condições abaixo:
 - Ser gerado um número aleatório entre 0 e 1 menor que o parâmetro G_p ;
 - O módulo do gradiente de F satisfizer $\|\partial F/\partial s_i\| < G_{maxG}$;
 - $\|\vec{v}\| \geq G_{minV}$ (essa condição será explicada a seguir).

onde $G_{threshold}$, G_p , G_{maxG} e G_{minV} são parâmetros de entrada do algoritmo e basta que uma das duas condições seja satisfeita para haver a mudança de modo, onde a segunda condição é, na verdade, um conjunto de três sub-condições que devem ser simultaneamente verdadeiras. A seguir, serão analisadas cada uma das condições acima.

A **primeira condição** suficiente, ou seja, $\|\partial F/\partial s_i\| < G_{threshold}$, fará com que o agente sempre retorne para o modo aleatório quando encontrar uma situação em que haja muito pouco ou nenhum evento ocorrendo em seu entorno.

O parâmetro $G_{threshold}$ deve ser ajustado com um valor pequeno de forma que os agentes só retornem ao modo aleatório devido a essa condição nos casos onde não há ou praticamente não há mais eventos ocorrendo em seu entorno.

Essa condição foi inserida com o objetivo de impedir que os agentes permaneçam no modo gradiente quando se depararem com uma situação onde não há mais eventos ocorrendo.

A **segunda condição** é, na verdade, um conjunto de 3 condições que, com o intuito de melhorar o entendimento, serão sempre denominadas de **sub-condições** neste trabalho.

A **primeira sub-condição** tem por objetivo diminuir a frequência com que os agentes deixam o modo gradiente mesmo que eles satisfaçam as outras duas sub-condições apresentadas a seguir . Note que esta sub-condição torna essa condição não determinística pois, mesmo que as duas sub-condições seguintes sejam satisfeitas, ainda assim poderá acontecer de o agente permanecer no modo gradiente. O parâmetro G_p deverá ser definido entre 0 e 1.

Essa sub-condição foi inserida com o objetivo de diminuir a frequência com a qual os agentes deixam o modo gradiente. Repare que as outras duas sub-condições, conforme será visto a seguir, restringem quais agentes estão “autorizados” a deixar o modo gradiente, enquanto que esta controla qual parcela desses agentes que, na média, voltará ao modo aleatório.

Agora, suponha um agente i que esteja em uma região cujo módulo de $\partial F/\partial s_i$ seja relativamente grande. Ora, se tal módulo é grande, isso significa que o agente está seguindo em direção a uma posição que fará com que o valor de F aumente relativamente bastante. Então por que esse agente deveria deixar de seguir o gradiente e voltar para o modo aleatório se ele está se movendo para tornar o valor de F consideravelmente maior? É melhor que apenas agentes já “estabilizados” possam voltar ao modo aleatório.

A **segunda sub-condição** determina justamente que apenas os agentes “estabilizados” poderão retornar ao modo aleatório. Repare que os parâmetros escolhidos devem ser tais que $G_{threshold} < G_{maxG}$ pois, caso contrário, sempre que

$$\|\partial F/\partial s_i\| < G_{maxG}$$

também seria verdade que

$$\|\partial F/\partial s_i\| < G_{maxG} \leq G_{threshold}$$

o que acarretaria que a segunda condição (que na verdade é composta de 3 sub-condições) seria inútil, pois o agente sempre deixaria o modo gradiente devido à primeira condição (módulo do gradiente menor que $G_{threshold}$).

Esta sub-condição foi inserida, também, pois, suponha um agente que acabou de entrar no modo gradiente porém não está em cima da sub-região onde há eventos ocorrendo (ele apenas está próximo o suficiente a ponto de os eventos estarem dentro de seu raio de sensoriamento), neste caso, tal agente poderia satisfazer as demais sub-condições e com isso deixar o modo gradiente com muita facilidade sem nem ao menos ter chegado ao local onde há eventos. Esta sub-condição visa impedir que tal situação ocorra pois, no caso mencionado, o módulo do gradiente de F seria grande o suficiente a ponto desta sub-condição não ser satisfeita.

A partir de agora, será falado a respeito da **terceira sub-condição**.

Voltando a olhar a Figura 4.6a, vamos considerar um momento em que os agentes no modo gradiente estão distribuídos uniformemente no local onde há eventos ocorrendo (o local de vermelho na figura).

Conforme já foi visto, é interessante que alguns agentes saiam do modo gradiente de tempos em tempos para explorar novas regiões onde possivelmente também haverá eventos ocorrendo (como é o caso da Figura 4.6b).

Dados os agentes distribuídos uniformemente na região em vermelho da Figura 4.6a e dado também que alguns deles deverão começar a andar aleatoriamente, quais são os mais apropriados a serem escolhidos para começar a executar o movimento Browniano? Os que estão no centro da região onde ocorrem eventos ou os que estão próximos à borda de tal região? Vamos analisar cada um dos casos.

Se um agente próximo ao centro do local onde estão ocorrendo eventos começar a andar aleatoriamente, tal agente continuará dentro do local onde os eventos estão ocorrendo por um bom tempo até que o mesmo consiga sair dela e começar a explorar regiões sem eventos, que é o seu objetivo ao parar de seguir o gradiente.

Na Figura 4.7 há um exemplo do que poderia acontecer com razoável probabilidade, onde a região mais escura é onde há incidência de eventos e as linhas representam os diversos passos do agente até que o mesmo consiga sair da região mais escura com eventos.

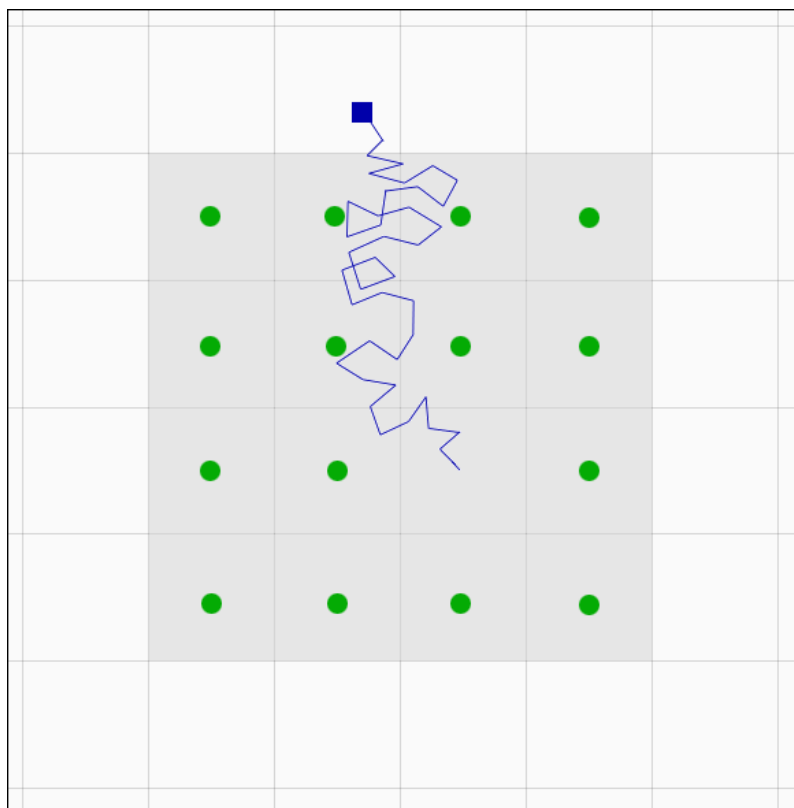


Figura 4.7: Exemplo do que acontece quando um agente de centro executa o movimento Browniano.

Vamos, agora, supor que o agente que começou a executar o movimento Browniano (ou seja, a andar aleatoriamente), foi um agente “de borda”. Nesse caso, o agente pode rapidamente se distanciar dos eventos ou pode permanecer andando perto dos mesmos. Na Figura 4.8, há três exemplos dos passos do agente após escolher trocar para o modo aleatório.

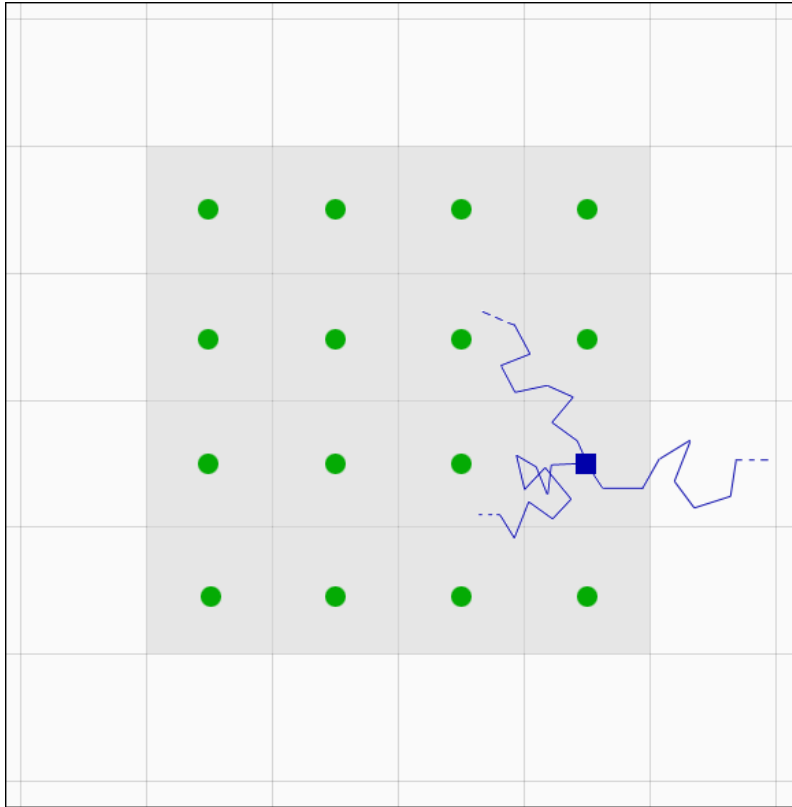


Figura 4.8: Possíveis passos do agente ao escolher o modo aleatório.

Dos três exemplos mostrados na Figura 4.8, em apenas um o agente assume o comportamento desejado, ou seja, afasta-se do local onde estão ocorrendo eventos. Sabe-se que, dadas as características do movimento Browniano, é relativamente improvável que o agente assumira esse comportamento desejado, ou seja, que o agente afaste-se rapidamente da região onde ocorrem eventos. Em muitos casos, o agente, ao invés de começar a explorar novos lugares, continuará fazendo o mesmo papel dos agentes que ainda estão no modo gradiente.

Com isso, conclui-se o seguinte:

- É desejável que apenas os agentes próximos à borda da região onde há eventos ocorrendo (regiões sombreadas nos exemplos acima) possam mudar para o modo aleatório;
- É desejável que, ao iniciar o movimento aleatório, os primeiros passos sejam para afastar o agente do local onde há eventos.

Seja N_i o conjunto dos vizinhos do agente i em relação ao raio de sensoriamento, ou seja, de todos os agentes cuja distância para o agente i é menor do que o tamanho do raio de sensoriamento dos agentes. Formalmente,

$$N_i = \{k : \|s_i - s_k\| \leq R_s\}.$$

Para cada agente a em N_i , vamos traçar um vetor que começa em s_a e termina em s_i . O vetor \vec{v} é definido como sendo a soma vetorial desses vetores. Na Figura 4.9 há um exemplo gráfico do cálculo de \vec{v} onde o vetor de linha mais grossa representa a soma (ou seja, representa \vec{v}), enquanto que os vetores de linha mais fina representam os de origem em s_a e término em s_i . O círculo representa o raio de sensoriamento de i .

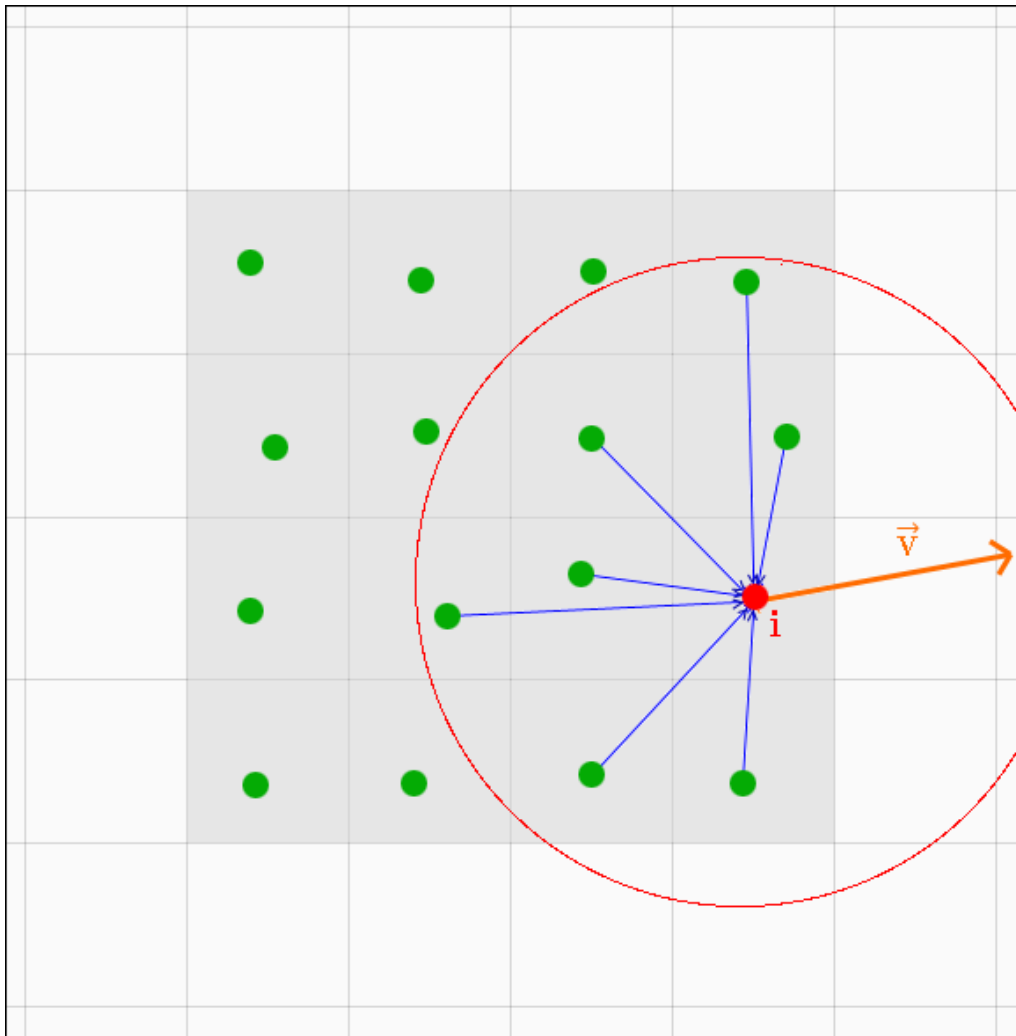


Figura 4.9: Exemplo do cálculo de \vec{v} . O vetor resultante está fora de escala.

Repare que \vec{v} indica a direção e sentido “para fora” da região onde há eventos ocorrendo. Logo, \vec{v} indica o sentido desejado que o agente deve seguir após entrar no modo aleatório.

Suponha, agora, o cálculo de \vec{v} para um agente i que esteja no centro. Se o agente está no centro, haverá agentes em volta de i por todos os lados, ao contrário de quando o agente está na borda, onde um dos lados não há agentes. Dessa forma, \vec{v} tende a ser menor já que os vetores opostos se anulam. Na Figura 4.10, há um exemplo deste caso, onde o vetor resultante \vec{v} é bem pequeno.

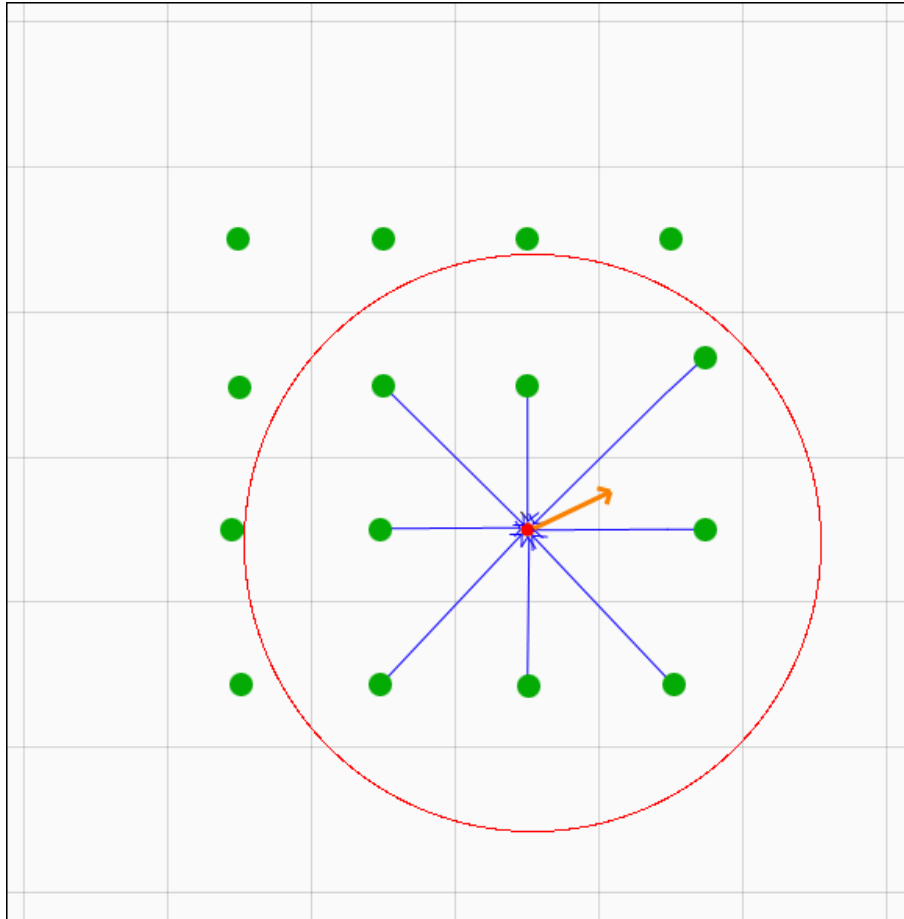


Figura 4.10: Exemplo do cálculo de \vec{v} para um agente no centro.

Com o objetivo de definir limites mínimos e máximos para o módulo do vetor \vec{v} , o mesmo, ao final do cálculo mostrado acima, é dividido pelo número de agentes que estão dentro do raio de sensoriamento do agente. Ou seja, o módulo de \vec{v} é dividido pelo número de vetores que foram utilizados no somatório vetorial que resultou em \vec{v} . Dessa forma, temos que $0 \leq \|\vec{v}\| \leq R_s$.

Vamos, agora, voltar para a terceira sub-condição. Para lembrar, tal sub-condição é $\|\vec{v}\| \geq G_{minV}$, ou seja, exige que o módulo de \vec{v} (após a normalização) seja maior do que um certo parâmetro de entrada do algoritmo.

Com essa sub-condição, o objetivo é que apenas os agentes localizados perto da borda da região com eventos sejam capazes de sair do modo gradiente e comecem a andar aleatoriamente. Não é garantido de forma alguma que todos os agentes na borda satisfaçam $\|\vec{v}\| \geq G_{minV}$, bem como também não é garantido que os que não

são da borda não satisfarão tal condição. Entretanto, para um valor apropriado de G_{minV} (que dependerá das características dos eventos e dos agentes), na maioria dos casos, só os agentes da borda satisfarão a desigualdade.

4.2.8 O terceiro modo de execução

Na subseção anterior, foi visto que é desejável que os agentes no centro da localidade sendo sensoriada permaneçam no modo gradiente, enquanto que apenas os da borda possam mudar para o modo aleatório.

Também foi visto que, quando um agente sai do modo gradiente, é desejável que o mesmo se distancie rapidamente do local onde estava pois, caso contrário, estaria fazendo o mesmo papel que estava antes, ou seja, continuaria sensoriando o mesmo local ao invés de explorar novos locais onde possivelmente há eventos ocorrendo. Esta subseção tratará exatamente de como fazer o agente ir para “longe” quando o mesmo sair do modo gradiente.

Além do modo aleatório e do modo gradiente, há também um terceiro modo chamado **modo de repulsão**. Sempre que um agente decide sair do modo gradiente, obrigatoriamente passa pelo modo de repulsão antes de ir para o modo aleatório. O objetivo do modo de repulsão, como o nome sugere, é repelir o agente para longe do local que ele estava.

O agente, ao entrar no modo de repulsão, permanecerá no mesmo durante exatamente S_r passos, onde S_r é um parâmetro de entrada do algoritmo. Após cada passo no modo de repulsão, assim como nos demais modos, o agente permanecerá um tempo parado.

Cada passo do agente no modo de repulsão tem tamanho S_a , ou seja, o mesmo tamanho de um passo no modo aleatório. Pode-se, inclusive, dizer que o comportamento do agente no modo de repulsão é idêntico ao comportamento no modo aleatório, exceto pela direção do movimento, conforme será visto a seguir, e pela condição para que o agente mude de modo (ao contrário do modo aleatório, no modo de repulsão o agente sai depois de exatos S_r passos).

Todos os S_r passos no modo de repulsão são executados exatamente em uma mesma direção e sentido, ou seja, ao final de S_r passos, o agente estará a exatas $S_r S_a$ unidades de distância da posição que ele estava ao entrar neste modo.

Relembrando, o agente pode sair do modo gradiente por 2 motivos: o módulo do gradiente é menor que $G_{threshold}$ (primeira condição) ou as três sub-condições são verdadeiras (segunda condição), onde uma delas é o módulo de \vec{v} ser maior do que G_{minV} .

A direção e o sentido escolhidos pelo agente para seguir durante os passos no modo de repulsão dependem da condição pela qual ele deixou o modo gradiente.

Se o agente deixou o modo gradiente devido à primeira condição

Se o agente chegou a uma situação em que o módulo de F era muito pequeno, na maioria dos casos (considerando que os parâmetros foram bem escolhidos), isso aconteceu devido ao encerramento de eventos naquele local. Por exemplo, suponha que o evento consista em cair uma gota de chuva e parou de chover. Como não haverá mais eventos sendo detectados, o módulo do gradiente de F se aproximará de zero. Na subseção a seguir, será explicado como é calculada a densidade de eventos baseando-se na amostra que se tem dos eventos.

Se não há mais eventos, o módulo do gradiente de F se aproximará de zero em todos os agentes que estavam sensoriando aquela localidade no modo gradiente. Neste caso, tem-se que todos aqueles agentes sairão do modo gradiente quase que simultaneamente.

Neste caso, com o objetivo de que todos esses agentes se dispersem rapidamente, cada um escolherá uma direção aleatória para seguir enquanto estiver no modo de repulsão.

Se o agente deixou o modo gradiente devido à segunda condição

Se o agente deixou o modo gradiente por ter satisfeito as 3 sub-condições, então ele já tem calculado \vec{v} , pois esse cálculo é necessário para que a terceira sub-condição possa ser avaliada.

O propósito de \vec{v} é que este indique uma direção que aponte para longe dos demais agentes que estão no modo gradiente. Ora, mas se \vec{v} indica uma direção e sentido que são opostas à massa de agentes naquela localidade, então, ao entrar no modo de repulsão, o agente seguirá a direção e sentido apontados por \vec{v} .

4.2.9 Calculando a densidade dinâmica de eventos

Em diversos momentos desta dissertação, foi visto que os agentes precisam conhecer a densidade instantânea de eventos para, por exemplo, calcular o gradiente de F , pois afinal os agentes precisam saber onde estão ocorrendo mais eventos para saberem qual o melhor lugar para se posicionar.

Implicitamente, há uma função que rege a densidade de eventos com o passar do tempo; entretanto, os agentes não têm acesso a ela. Os agentes só tem acesso à amostra de eventos que foram por eles sensorizados diretamente ou que foram sensorizados por algum outro agente e que chegou até eles através da rede de comunicação.

A questão a ser respondida a seguir é: como calcular a densidade instantânea ϕ de eventos tendo como informação apenas a amostra de eventos que foram sensorizados?

Primeiramente, vamos discretizar o domínio (região), cujas dimensões são R_L e R_A , respectivamente na largura e na altura, em D_L e D_A partes na largura e na

altura, novamente respectivamente. Cada uma dessas $D_L D_A$ partes será a partir de agora chamada de **subdivisão**. Cada subdivisão tem largura R_L/D_L e altura R_A/D_A .

Então como o agente pode calcular a densidade $\phi(i, j, t)$ para a subdivisão (i, j) no tempo t ? Uma possibilidade é que o valor de $\phi(i, j, t)$ seja o número total de eventos dentro desta subdivisão dos quais o agente tomou conhecimento (através do sensoriamento direto ou por ter recebido uma mensagem informando).

A forma para calcular ϕ proposta no parágrafo anterior será dinâmica, pois quando o agente sensoriar novos eventos, os mesmos serão incluídos no cálculo, alterando o valor de ϕ . Entretanto, a mesma contempla, também, para cada subdivisão (i, j) , os eventos que ocorreram há muito tempo atrás.

Veja a Figura 4.11. Em 4.11a, há eventos ocorrendo na localidade A e, em 4.11b, há eventos ocorrendo na localidade B que é totalmente distinta da localidade A. Ora, se ϕ for calculado conforme já foi proposto, após 40 horas depois do começo do experimento, ainda haverá agentes seguindo o gradiente na localidade A, pois eles entenderão que em A ainda há eventos acontecendo visto que todos os eventos (desde o começo do experimento) estão sendo contabilizados ao calcular ϕ , o que não é mais verdade.

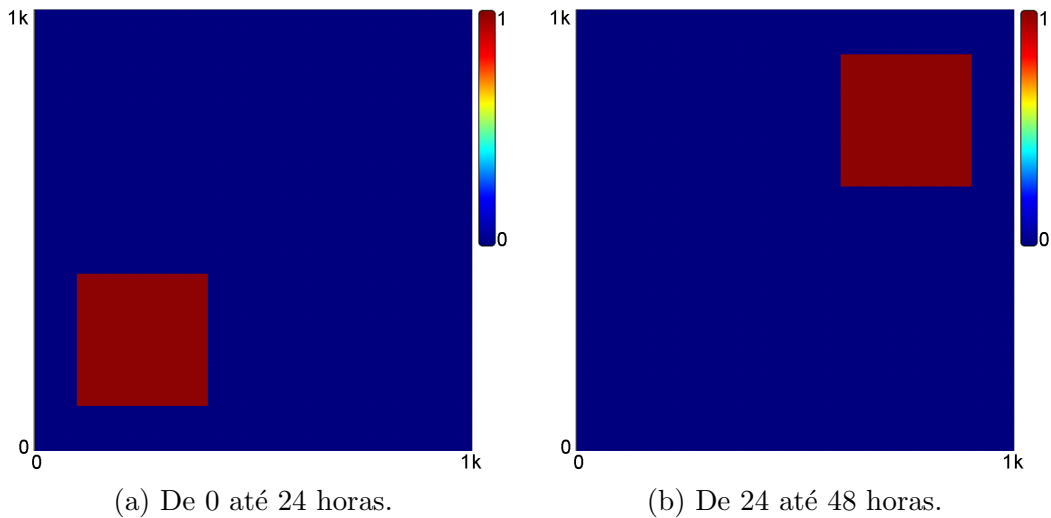


Figura 4.11: Densidade de eventos dinâmica.

Para aumentar a precisão de ϕ , ou seja, para que ϕ represente melhor a densidade instantânea (ao invés de representar a densidade histórica), em seu cálculo só serão contabilizados os eventos que ocorreram no intervalo $[t - T_{int}, t]$ onde t é o tempo atual e T_{int} é um parâmetro do algoritmo que indica o tempo de interesse pelos eventos. Em outras palavras, a densidade ϕ será calculada através apenas dos eventos que ocorreram nas últimas T_{int} unidades de tempo.

4.3 Parâmetros

Para executar uma simulação, há diversos parâmetros cujos valores mais apropriados dependem das características do agente e dos eventos que estão sendo simulados.

Não há uma forma automática de escolher otimamente todos os parâmetros. O método utilizado para a escolha dos mesmos consiste em começar utilizando valores aproximados e, através da inspeção dos resultados (incluindo a inspeção visual do movimento dos agentes na região), realizar um aprimoramento do que foi previamente utilizado como entrada do algoritmo, até que o resultado esteja satisfatório, onde a definição de satisfatório varia de acordo com cada caso.

Apesar de não ser o caso de todos, alguns parâmetros podem ser trivialmente escolhidos. É o caso dos parâmetros que dependem das características dos eventos ou das características dos agentes como, por exemplo, o tempo de visibilidade do evento; este é um parâmetro que não pode ser variado pois é uma característica física do evento que está sendo modelado.

Abaixo serão explicados cada um dos parâmetros utilizados como entrada do experimento e, quando aplicável, quais implicações terá a simulação quando tal parâmetro for ajustado em um valor muito grande ou em um valor muito pequeno.

É importante ressaltar que todos os parâmetros relacionados à distância possuem a mesma unidade (metros, centímetros, km, etc), podendo ser convencionada qualquer uma delas, desde que todos os parâmetros possuam a mesma unidade. O mesmo vale para os parâmetros de tempo.

4.3.1 Largura e altura da região

A região do experimento, ou seja, o plano por onde os agentes têm liberdade de andar e que coincide com o local onde os eventos podem acontecer, consiste em um retângulo de dimensões R_L e R_A para a largura e altura, respectivamente.

4.3.2 Número de subdivisões da largura e altura

Número de subdivisões feitas na largura e na altura, denotados por D_L e D_A , respectivamente, de forma a discretizar a região onde ocorrem os eventos para, assim, ser possível calcular a densidade de eventos ϕ , ou seja, calcular o número de eventos que ocorreram em cada uma dessas $D_L D_A$ subdivisões.

4.3.3 Tempo simulado

Denotado por T , o tempo simulado consiste, como o nome sugere, em quantas unidades de tempo serão simuladas pelo simulador construído. Após T unidades de tempo terem sido simuladas, a simulação termina.

4.3.4 Tempo de sensoriamento

O tempo de sensoriamento consiste no tempo que o agente permanecerá imóvel antes de executar algum movimento, qualquer que seja o seu modo de operação (aleatório, gradiente ou repulsão).

Este parâmetro é uma variável aleatória normal com média de m unidades de tempo e desvio padrão σ . Ou seja, o tempo que o agente permanecerá parado fazendo o sensoriamento de seu entorno será dado por $N(m, \sigma)$.

Ajustando o parâmetro

Se este parâmetro for grande, os agentes farão menos movimentos durante a simulação (em média, T/m passos) de forma que eles tendem a caminhar por uma região menor.

Se for pequeno, o agente fará muitos movimentos, desperdiçando energia.

4.3.5 Número mínimo de passos no modo aleatório

Denotado por A_{minS} , é o número mínimo de passos aleatórios que o agente deverá executar toda vez que entrar neste modo. Enquanto não completar A_{minS} passos, o agente obrigatoriamente deverá permanecer executando movimentos aleatórios para só depois ser possível mudar para o modo gradiente.

Ajustando o parâmetro

Se este parâmetro for grande, o agente terá mais dificuldade para entrar no modo gradiente, pois será necessário um tempo maior no modo aleatório.

Se for pequeno, ao sair do modo gradiente, é possível que o agente retorne ao modo gradiente muito rapidamente antes de explorar outras localidades.

4.3.6 Gradiente mínimo de F pra sair do modo aleatório

Conforme visto, para que um agente saia do modo aleatório e entre no modo gradiente, é necessário que o módulo do gradiente de F esteja acima de um certo valor. O valor mínimo a tal mudança de modo será denotado por $A_{threshold}$.

Ajustando o parâmetro

Se for grande, o agente só conseguirá entrar no modo gradiente em alguns poucos casos onde seja calculado um gradiente grande. Se for muito grande, pode ser até mesmo que o módulo do gradiente de F nunca atinja tal valor.

Se for pequeno (porém maior que zero), o agente entrará com mais facilidade no modo gradiente o que, em muitos casos, não é um problema desde que seja garantido que ele só entrará no modo gradiente quando seu módulo for maior que zero (caso contrário, ficaria imóvel).

4.3.7 Gradiente mínimo de F necessário para permanecer no modo gradiente

Denotado por $G_{threshold}$, é o valor mínimo que o módulo do gradiente de F deve ter para que o agente possa permanecer no modo gradiente. Se em algum momento o agente calcular um valor para $\|\partial F/\partial s_i\|$ inferior a $G_{threshold}$, imediatamente o mesmo deixará de se movimentar seguindo o gradiente.

Ajustando o parâmetro

Se for grande, o agente sairá do modo gradiente muito facilmente. Se for pequeno, será mais difícil de sair (por conta dessa condição).

A regra geral, neste caso, é manter um valor bem pequeno de forma que o agente só saia do modo gradiente devido a esta condição caso o gradiente calculado atinja um valor irrisório (zero ou praticamente zero).

4.3.8 Aleatoriedade para sair do modo gradiente

A primeira das 3 sub-condições para sair do modo gradiente é que seja escolhido um número aleatório entre 0 e 1 e que tal número seja menor que este parâmetro, que será denotado por G_p . Ou seja, este parâmetro G_p indica a probabilidade de que, dado que as 2 sub-condições explicadas a seguir foram satisfeitas, qual a probabilidade de o agente sair do modo gradiente?

Ajustando o parâmetro

Se for muito grande (próximo de 1), bastará que as outras duas condições (G_{minV} e G_{maxG}) sejam satisfeitas para que o agente saia do modo gradiente.

Se for muito pequeno (próximo de 0), mesmo com as outras duas condições sendo satisfeitas, o agente não conseguirá sair do modo gradiente.

4.3.9 Gradiente máximo de F para sair do modo gradiente

A segunda das 3 sub-condições para sair do modo gradiente, lembrando, é que o módulo do gradiente de F satisfaça $\|\partial F/\partial s_i\| < G_{maxG}$, onde G_{maxG} é justamente esta parâmetro.

Ajustando o parâmetro

Se for grande, será possível que um agente que está se movimentando rapidamente saia do modo gradiente, o que não é interessante pois, se o gradiente é grande, é sinal que o agente está caminhando para um local de grande interesse.

Se for pequeno, será muito difícil sair do modo gradiente. A regra geral é escolher um valor intermediário.

4.3.10 Módulo mínimo de \vec{v}

A terceira e última sub-condição consiste em o módulo do vetor \vec{v} ser maior do que um certo valor. Tal valor será denotado por G_{minV} , de forma que \vec{v} deverá satisfazer $\|\vec{v}\| \geq G_{minV}$ para que o agente deixe o modo gradiente devido à segunda condição (que consiste em 3 sub-condições).

Ajustando o parâmetro

Se for grande, apenas alguns poucos agentes na borda conseguirão satisfazer esse parâmetro.

Se for pequeno, qualquer agente conseguirá satisfazer, inclusive os que estão no meio do local onde estão ocorrendo eventos.

Repare que o maior valor possível para este parâmetro é R_s .

4.3.11 Número de passos no modo de repulsão

O parâmetro S_r indica o número exato de passos que o agente executará enquanto estiver no modo de repulsão.

Ajustando o parâmetro

Se for muito grande, ao sair do modo gradiente, o agente irá ser repellido para muito longe, enquanto que, se pequeno, o agente continuará muito próximo do local que estava anteriormente quando voltar a andar aleatoriamente.

4.3.12 Número de agentes

O parâmetro N indica o número de agentes que serão utilizados na simulação.

Ajustando o parâmetro

Quanto maior o número de agentes, maior será a qualidade do resultado, pois haverá mais agentes fazendo o sensoriamento da região, de forma que a probabilidade de um

evento ser detectado e que essa informação se propague pela rede de comunicação até atingir todos os agentes será maior.

Em contrapartida, quanto menor esse número, menor o custo com a aquisição dos mesmos (no mundo físico apenas). A regra geral é escolher o menor número de agentes tal que a qualidade do resultado final seja satisfatória (onde o resultado ser satisfatório ou não dependerá da aplicação prática).

4.3.13 Tamanho do passo aleatório

Denotado por S_a , este parâmetro indica o comprimento de um passo no modo aleatório.

Ajustando o parâmetro

Quanto maior, mais energia será gasta com a movimentação do agente.

Quanto menor, mais dificuldade o agente terá de explorar regiões mais distantes pois, na média, será necessário um número maior de passos até que ele consiga se afastar de certa posição.

4.3.14 Tamanho do passo de referência no modo gradiente

O parâmetro S_g é uma referência à distância que um agente no modo gradiente percorrerá em um passo quando estiver no modo gradiente. Lembrando que o tamanho real do passo a ser executado será

$$S_g \left\| \frac{\partial F}{\partial s_i} \right\|.$$

Ajustando o parâmetro

Como o tamanho de um passo no modo gradiente é calculado da maneira explicada acima, quanto maior este parâmetro, maior o passo do agente no modo gradiente. Se for grande, o agente sempre executará movimentos “bruscos” de forma que nunca conseguirá chegar na posição ótima (que precisa de movimentos pequenos para que seja possível alcançá-la).

Se for pequeno, o agente demorará muito tempo até conseguir chegar em determinada posição apontada pela direção do vetor gradiente.

O valor especificado para este parâmetro deve ser inversamente proporcional à média dos módulos dos gradientes calculados durante a simulação, que dependem diretamente do número de eventos e de T_{int} .

4.3.15 Raio de sensoriamento

Cada agente é capaz de sensoriar eventos que ocorreram até R_s unidades de distância dele. Entretanto, quando mais longe, menor será a probabilidade de que o agente consiga detectar o evento.

Ajustando o parâmetro

Quanto maior, melhor a capacidade de sensoriamento do agente. Logo, melhor será a qualidade do experimento.

Quanto menor, menor será o consumo de energia gasto no sensoriamento. A regra geral é utilizar o maior valor que seja fisicamente viável.

4.3.16 Raio de comunicação

Toda mensagem enviada por um agente poderá ser recebida pelos agentes que estiverem em até R_c unidades de distância dele.

Ajustando o parâmetro

Quanto maior, melhor a capacidade do agente de se comunicar com os demais e, com isso, maiores as chances de, dado que um evento foi detectado por algum agente, todos os demais agentes tomarem conhecimento da existência desse evento, melhorando a qualidade do experimento.

Quanto menor, menor será o consumo de energia gasto na comunicação. A regra geral é utilizar o maior valor que seja fisicamente viável.

4.3.17 Probabilidade de recebimento de mensagem

Toda mensagem enviada por um agente poderá ser recebida pelos agentes que estiverem em até R_c unidades de distância dele com probabilidade P_c .

Ajustando o parâmetro

Esse parâmetro está diretamente relacionado a R_c , de forma que aumentá-lo ou diminuí-lo terá as mesmas consequências vistas anteriormente.

4.3.18 Posição inicial

Posição dos agentes no começo da simulação. Pode-se posicionar cada agente em uma posição aleatória ou todos os agentes podem começar posicionados em um ponto específico (também passado como parâmetro).

4.3.19 Tempo de visibilidade

Tempo de visibilidade do evento. Após ter ocorrido, cada evento ainda poderá ser detectado por um agente por mais T_{vis} unidades de tempo. Esse parâmetro depende da natureza do evento que está sendo modelado.

4.3.20 Tempo de interesse

Cada agente constrói sua visão da densidade de eventos no domínio através da sua lista particular de eventos detectados (seja o evento detectado por ele próprio ou recebido por mensagem). Eventos que ocorreram há mais de T_{int} são desprezados no cálculo da densidade, pois, de acordo com o problema que está sendo modelado, pode ser que considerar apenas os eventos nas últimas T_{int} unidades de tempo faça com que haja uma precisão melhor da densidade instantânea atual.

Ajustando o parâmetro

Quanto maior esse parâmetro, maior será a precisão da densidade calculada, pois haverá uma amostra maior (considerando um intervalo de tempo maior, haverá mais eventos na amostra em relação a um intervalo menor). Em contrapartida, com um T_{int} muito grande, eventos que ocorreram há muito tempo atrás também estarão sendo considerados ao calcular a densidade; porém estes eventos muito antigos podem não mais representar o cenário atual e, assim, diminuam a precisão da densidade calculada.

A escolha deste parâmetro depende fundamentalmente da natureza dos eventos que estão sendo considerados.

4.4 A função F

A função F retorna, dado que um evento qualquer ocorreu (de acordo com a densidade ϕ), qual a probabilidade de que pelo menos um agente detecte tal evento. Quando um agente está no modo gradiente, cada passo do mesmo é regido pela direção do gradiente da função F em relação à posição do agente. Esta seção dedica-se a explicar esta função.

Relembrando, a probabilidade de o agente i na posição s_i detectar um evento na posição q é

$$p_i(q) = \begin{cases} \left(\frac{d_i - R_s}{R_s}\right)^2 & \text{se } d_i \leq R_s \\ 0 & \text{c.c.} \end{cases} \quad (4.1)$$

onde $d_i = \|q - s_i\|$.

Dados N agentes nas posições $s = (s_1, \dots, s_N)$, a probabilidade de que um evento que ocorreu na posição q seja detectado por algum agente é

$$P(s|q) = 1 - \prod_{k=1}^N [1 - p_k(q)]. \quad (4.2)$$

Seja a função $\phi(q)$ que representa a densidade de ocorrência de eventos na posição q , onde $q \in Q$, $Q \subset \mathbb{R}^2$ é a região onde ocorre o experimento, $\phi(q) \geq 0$ para todo $q \in Q$ e $\int_Q \phi(q) dq < \infty$,

A probabilidade de que, dado que um evento qualquer ocorreu em Q de acordo com a densidade ϕ , ele tenha sido detectado é proporcional a:

$$F(s) \propto \int_Q \phi(q) P(q, s) dq. \quad (4.3)$$

E esta é a função objetivo a ser maximizada. Foi utilizado o símbolo de proporcionalidade pois é possível que $\int_Q \phi(q) dq > 1$. Juntando as equações (4.2) e (4.3), a função F pode ser reescrita conforme a equação (4.4) ou conforme a equação (4.5) mostradas a seguir, onde foi utilizado um sinal de igualdade visto que o valor absoluto de F não é relevante, apenas o valor de seu gradiente.

$$F(s) = \int_Q \phi(q) \left[1 - \prod_{k=1}^N [1 - p_k(q)] \right] dq, \quad (4.4)$$

$$F(s) = \int_Q \phi(q) dq - \int_Q \phi(q) \prod_{k=1}^N [1 - p_k(q)] dq. \quad (4.5)$$

Agora, vamos calcular a derivada parcial de F em relação a s_i , ou seja, em relação à posição do agente i :

$$\frac{\partial F}{\partial s_i} = - \int_Q \phi(q) \frac{\partial \left[\prod_{k=1}^N [1 - p_k(q)] \right]}{\partial s_i} dq \quad (4.6)$$

$$\frac{\partial F}{\partial s_i} = - \int_Q \phi(q) (1 - p_1(q)) \dots \frac{\partial(1 - p_i(q))}{\partial s_i} \dots (1 - p_N(q)) dq. \quad (4.7)$$

Como $p_i(q) = 0$ quando $d_i > R_s$ (raio de sensoriamento), podemos mudar o domínio da integral para

$$\frac{\partial F}{\partial s_i} = - \int_{Q_i} \phi(q) (1 - p_1(q)) \dots \frac{\partial(1 - p_i(q))}{\partial s_i} \dots (1 - p_N(q)) dq \quad (4.8)$$

onde $Q_i = \{q \in Q : \|q - s_i\| \leq R_s\}$. Como $p_k(q) = 0$ quando $Q_i \cap Q_k = \emptyset$, visto que o domínio da integral é Q_i , temos que

$$\frac{\partial F}{\partial s_i} = - \int_{Q_i} \phi(q) \left[\frac{\partial(1 - p_i(q))}{\partial s_i} \right] \left[\prod_{k \in N'_i} (1 - p_k(q)) \right] dq, \quad (4.9)$$

onde N'_i é o conjunto dos agentes capazes de sensoriar pelo menos um ponto em comum com o agente i e definido por $N'_i = \{k : \|s_i - s_k\| \leq 2R_s, i \neq k\}$.

A partir de agora, vamos nos limitar à derivada $\partial p_i(q)/\partial s_i$. Primeiramente, temos que:

$$p_i(q) = \left(\frac{d_i - R_s}{R_s} \right)^2 = \frac{d_i^2}{R_s^2} - \frac{2d_i}{R_s} + 1. \quad (4.10)$$

Lembrando que $d_i = \|q - s_i\|$ onde $q = (q_x, q_y)$ e $s_i = (s_{ix}, s_{iy})$, temos:

$$p_i(q) = \frac{(s_{ix} - q_x)^2 + (s_{iy} - q_y)^2}{R_s^2} - \frac{2\sqrt{(s_{ix} - q_x)^2 + (s_{iy} - q_y)^2}}{R_s} + 1. \quad (4.11)$$

Vamos, agora, derivar $p_i(q)$ em relação a s_{ix} para, com isso, obtermos o valor da componente x do vetor gradiente. O cálculo da componente y é análogo.

$$\frac{\partial p_i}{\partial s_{ix}} = \frac{2(s_{ix} - q_x)}{R_s^2} - \frac{2}{R_s} \frac{1}{2\sqrt{(s_{ix} - q_x)^2 + (s_{iy} - q_y)^2}} 2(s_{ix} - q_x) \quad (4.12)$$

$$\frac{\partial p_i}{\partial s_{ix}} = \frac{2}{R_s} (s_{ix} - q_x) \left(\frac{1}{R_s} - \frac{1}{\sqrt{(s_{ix} - q_x)^2 + (s_{iy} - q_y)^2}} \right) \quad (4.13)$$

Analogamente, tem-se que

$$\frac{\partial p_i}{\partial s_{iy}} = \frac{2}{R_s} (s_{iy} - q_y) \left(\frac{1}{R_s} - \frac{1}{\sqrt{(s_{ix} - q_x)^2 + (s_{iy} - q_y)^2}} \right) \quad (4.14)$$

Aplicando 4.13 a 4.9, tem-se

$$\frac{\partial F}{\partial s_{ix}} = - \int_{Q_i} \phi(q) \left[-\frac{2}{R_s} (s_{ix} - q_x) \left(\frac{1}{R_s} - \frac{1}{\sqrt{(s_{ix} - q_x)^2 + (s_{iy} - q_y)^2}} \right) \right] \left[\prod_{k \in N'_i} (1 - p_k(q)) \right] dq \quad (4.15)$$

Eliminando o sinal de menos, temos que:

$$\frac{\partial F}{\partial s_{ix}} = \int_{Q_i} \phi(q) \left[\frac{2}{R_s} (s_{ix} - q_x) \left(\frac{1}{R_s} - \frac{1}{\sqrt{(s_{ix} - q_x)^2 + (s_{iy} - q_y)^2}} \right) \right] \left[\prod_{k \in N'_i} (1 - p_k(q)) \right] dq \quad (4.16)$$

Aplicando (4.14) a (4.9) e eliminando o sinal de menos, tem-se que:

$$\frac{\partial F}{\partial s_{iy}} = \int_{Q_i} \phi(q) \left[\frac{2}{R_s} (s_{iy} - q_y) \left(\frac{1}{R_s} - \frac{1}{\sqrt{(s_{ix} - q_x)^2 + (s_{iy} - q_y)^2}} \right) \right] \left[\prod_{k \in N'_i} (1 - p_k(q)) \right] dq. \quad (4.17)$$

Lembrando que, por ser contínua, o agente não é capaz de calcular ϕ . Porém, o agente tem calculada a função ϕ' , onde $\phi'(i, j)$ representa o número de eventos que ocorreram na subdivisão (i, j) dentro do tempo de interesse. Dessa forma, troca-se a integral por dois somatórios para aproximarmos o valor da derivada parcial:

$$\frac{\partial F}{\partial s_{ix}} \approx \sum_x \sum_y \phi'(x, y) \left[\frac{2}{R_s} (s_{ix} - q_x) \left(\frac{1}{R_s} - \frac{1}{\sqrt{(s_{ix} - q_x)^2 + (s_{iy} - q_y)^2}} \right) \right] \left[\prod_{k \in N'_i} (1 - p_k(q)) \right]. \quad (4.18)$$

Onde o ponto $q = (q_x, q_y)$ considerado está exatamente no centro geométrico da subdivisão (x, y) . O caso para $\partial F / \partial s_{iy}$ é análogo.

4.5 Algoritmo

O algoritmo 1 é executado uma única vez para inicialização das variáveis internas dos agentes antes de começar o sensoriamento.

Algoritmo 1: Inicialização do agente

```
1 início
2   eventosConhecidos  $\leftarrow \emptyset$ ;
3   modoOperacao  $\leftarrow$  “aleatorio”;
4   numPassosEstadoAtual  $\leftarrow 0$ ;
5   anguloRepulsao  $\leftarrow 0$ ;
6   for  $i \leftarrow 1$  to  $N$  do
7     [ posicaoAgentes[  $i$  ]  $\leftarrow (-\infty, -\infty)$ ;
8     permanecer imóvel por  $N(m, \sigma)$ ;
9 fim
```

O algoritmo 2 é executado toda vez que um agente recebe uma mensagem de algum de seus vizinhos.

Algoritmo 2: Ao receber uma mensagem

```
1 início
2   msg  $\leftarrow$  mensagem recebida;
3   se  $msg \notin msgsRecebidas$  então
4     [ msgsRecebidas.add(msg);
5     [ posicaoAgentes[ msg.autor.id ]  $\leftarrow$  msg.autor.posicao;
6     for  $evt \in msg.eventos$  do
7       [ se  $evt \notin eventosConhecidos$  então
8         [ [ eventosConhecidos.add(evt);
9         [ propaga msg aos vizinhos;
10 fim
```

A seguir, o que acontece quando o agente percebe a presença de um evento é implementado no algoritmo 3.

Algoritmo 3: Ao sensoriar um evento

```
1 início
2   evento  $\leftarrow$  evento sensoriado;
3   se evento  $\notin eventosConhecidos$  então
4     [ evtSensoreados.add(evento);
5 fim
```

O algoritmo 4 é executado toda vez que chegar o momento em que o agente

deverá executar um movimento.

Algoritmo 4: Quando estiver na hora de o agente se mover

```
1 início
   /* Armazena os eventos sensorizados na variável eventosConhecidos */
2 for evt ∈ evtsSensorizados do
3   se evt ∉ eventosConhecidos então
4     eventosConhecidos.add(evt);
   /* Verifica se deve alterar o modo de execução */
5 if modoOperacao = “repulsao” AND numPassosEstadoAtual ≥ Sr then
6   modoOperacao ← “aleatorio”;
7   numPassosEstadoAtual ← 0;
8 else if modoOperacao = “gradiente” AND getModuloGradienteDeF()
   < Gthreshold then
9   modoOperacao ← “repulsao”;
10  numPassosEstadoAtual ← 0;
11  anguloRepulsao ← rand(0, 360);
12 else if modoOperacao = “gradiente” AND rand(0,1) < Gp AND
   getModuloGradienteDeF() < GmaxG AND getModuloVetorV() < GminV then
13   modoOperacao ← “repulsao”;
14   numPassosEstadoAtual ← 0;
15   anguloRepulsao ← mesmo angulo de  $\vec{v}$ ;
16 if modoOperacao = “aleatorio” AND numPassosEstadoAtual ≥ AminS AND
   getModuloGradienteDeF() > Athreshold then
17   modoOperacao ← “gradiente”;
18   numPassosEstadoAtual ← 0;
   /* Executa o movimento apropriado e limpa evtsSensorizados */
19 numPassosEstadoAtual = numPassosEstadoAtual + 1;
20 if modoOperacao = “repulsao” then
21   anda Sa unidades no ângulo anguloRepulsao;
22 else if modoOperacao = “gradiente” then
23    $\vec{G}$  ← getVetorGradienteDeF();
24   anda Sg || $\vec{G}$ || unidades na mesma direção de  $\vec{G}$ ;
25 else
26   anda Sa unidades em uma direção arbitrária;
27 evtsSensorizados ← ∅;
28 msg.autor.id ← id deste agente;
29 msg.autor.posicao ← posição atual deste agente;
30 msg.eventos ← evtsSensorizados;
31 propaga msg aos vizinhos;
32 permanece imóvel por N(m, σ) unidades de tempo;
33 fim
```


A seguir, algumas observações a respeito deste último algoritmo apresentado que é executado toda vez que um agente vai se mover.

A função *getVetorGradienteDeF()* calcula, primeiramente, a densidade local ϕ de eventos considerando os eventos contidos na variável *eventosConhecidos*, tal que o evento tenha ocorrido há no máximo T_{int} unidades de tempo. Em seguida, a função calcula $\|\partial F/\partial s_i\|$ (que, lembrando, depende da densidade de eventos e onde s_i é a posição do agente na região) e retorna o valor que foi calculado (um par ordenado (x, y) representando um vetor).

A função *getModuloGradienteDeF()* retorna o módulo do vetor retornado pela função *getVetorGradienteDeF()*.

A função *getModuloVetorV()* começa tomando a posição de todos os agentes de acordo com *posicaoAgentes* tal que a distância para sua localização atual é no máximo R_s . Para cada uma destas posições, a função traça um vetor dela para a posição atual do agente e, em seguida, faz o somatório. O valor retornado será o módulo deste somatório dividido pelo número de vetores que foram utilizados no cálculo.

Capítulo 5

Experimentos

Neste capítulo, serão mostrados alguns experimentos que são bastante representativos pois possuem eventos com características (tempo de visibilidade) e padrões de ocorrência (densidade) diversos. Cada experimento está em uma seção própria e cada comportamento analisado foi baseado em uma única execução.

Para cada experimento, começamos explicando um exemplo do mundo real do que poderia estar sendo simulado. Em seguida, são mostrados os parâmetros que foram utilizados e, por último, os resultados obtidos.

Para cada experimento, no final da seção, há também um endereço de uma página na internet onde é possível acessar os resultados completos, incluindo gráficos interativos da movimentação dos agentes com o passar do tempo.

Por uma questão de simplicidade, todos os experimentos presentes nesta dissertação possuem alguns valores padrões para os parâmetros que serão utilizados exceto quando for especificado ao contrário. São eles:

Largura e altura da região

Todos os experimentos são executados baseando-se em uma região com 1000 unidades de largura e 1000 unidades de altura.

Número de subdivisões da largura e altura

Como forma de discretizá-las, as regiões de todos os experimentos são divididas em 100 partes tanto na altura quanto na largura, de forma que cada subdivisão terá tamanho 10x10 (ou seja, $1000/100 = 10$ tanto na largura quanto na altura).

Comunicação

A comunicação em todos os experimentos será feita através de um raio de comunicação $R_c = 200$ e probabilidade de sucesso no recebimento de $P_c = 90\%$.

Posição inicial dos agentes

A posição inicial dos agentes será aleatória.

Tamanho dos passos e número de passos na repulsão

O tamanho do passo quando o agente está no modo aleatório (e também no modo de repulsão) é $S_a = 25$ metros.

No modo gradiente, o tamanho do passo será o módulo do gradiente de F multiplicado por $S_g = 25$ metros.

Quando o agente entrar no modo repulsão, o mesmo executará 10 passos antes de sair do mesmo.

5.1 Chuva

Este experimento consiste na simulação de uma faixa vertical de chuva que começa do lado esquerdo e vai se deslocando com velocidade uniforme para a direita com o passar do tempo, até que tenha percorrido todo o eixo horizontal.

Cada gota de chuva que cai no solo é considerada um evento. Dessa forma, os eventos são instantâneos: ou o agente o detecta no momento de sua ocorrência ou não será mais capaz de detectá-lo.

5.1.1 Parâmetros

Eventos e duração

Neste experimento, durante 90k minutos (tempo simulado no experimento) há uma faixa vertical de largura 100 conforme a Figura 5.1a, que vai se deslocando para a direita com velocidade uniforme até atingir a extremidade direita da região (Figura 5.1b).

Foram gerados 900k eventos para este experimento, cujo tempo de visibilidade é $T_{vis} = 0$ e cujo tempo de interesse (para calcular a densidade de eventos) é $T_{int} = 1000$ pois este valor permite uma boa precisão sem que, para isso, seja necessário olhar os eventos muito antigos.

O tempo de visibilidade nulo (ou próximo a 0) define uma primeira classe de problemas que consiste justamente em um tempo de visibilidade muito pequeno.

Número e características dos agentes

Temos disponíveis para executar essa simulação 50 agentes cujo raio de sensoria-mento é de 100 metros (lembrando que todos os agentes são idênticos).

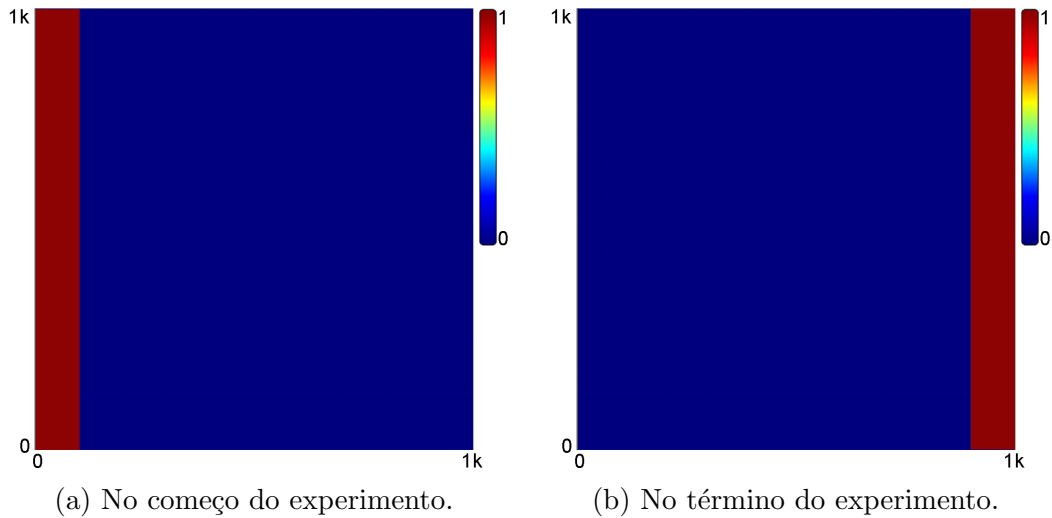


Figura 5.1: Densidade de eventos idealizada. Os eventos reais utilizados estão disponíveis na página do experimento.

Comportamento dos agentes

Para conseguirmos uma análise melhor do desempenho do algoritmo proposto nesta dissertação, foram analisados 3 comportamentos distintos para os agentes. São eles:

- **Os agentes operam no modo aleatório durante toda a simulação.** Este é o cenário de controle onde não há aplicação do algoritmo. Os parâmetros foram configurados de forma que o agente nunca deixe o modo aleatório;
- **Alguns agentes operam no modo gradiente e alguns, no modo aleatório.** Este caso é misto e os parâmetros utilizados estão especificados na página deste experimento descrita a seguir. Ao entrar no modo gradiente, alguns agentes voltam ao modo aleatório com certa frequência;
- **Todos os agentes operam no modo gradiente sempre que possível.** Como sabemos que os agentes seguirão sempre uma faixa vertical conforme a Figura 5.1, sempre que um agente entrar no modo gradiente, ficará neste modo para sempre, ou seja, ele não voltará a movimentar-se aleatoriamente à procura de novos locais onde possam ter surgido eventos, pois sabemos que isso nunca acontecerá (mas que nem sempre poderemos garantir).

Em todos os 3 casos acima, os agentes sempre escolherão o tempo em que permanecerão imóveis através de uma distribuição normal $N(10, 1)$.

Tabela de parâmetros

Na Tabela 5.1, são apresentados todos os parâmetros utilizados neste experimento.

Parâmetro	Comportamento		
	Aleatório	Misto	Gradiente
$R_L \times R_A$	1000x1000	1000x1000	1000x1000
$D_L \times D_A$	100x100	100x100	100x100
T	90000	90000	90000
$N(m, \sigma)$	$N(10, 1)$	$N(10, 1)$	$N(10, 1)$
A_{minS}	9999	50	0
$A_{threshold}$	9999	0.01	0.001
$G_{threshold}$	9999	0.0001	0.001
G_p	100%	10%	0
G_{maxG}	9999	0.05	0
G_{minV}	0	25	9999
S_r	10	10	10
N	50	50	50
S_a	25	25	25
S_g	25	25	25
R_s	100	100	100
R_c	200	200	200
P_c	90%	90%	90%
T_{vis}	0	0	0
T_{int}	1000	1000	1000

Tabela 5.1: Chuva: Tabela de parâmetros.

5.1.2 Resultados

Eventos sensoriados por pelo menos um agente

Na Tabela 5.2 a seguir, para cada um dos 3 comportamentos testados especificados acima, são mostradas as parcelas de eventos (dentre todos os eventos que ocorreram durante o experimento) onde pelo menos um agente foi capaz de sensoriá-los.

Mais detalhes sobre essa forma de mensurar os resultados está descrita na Seção 3.3.1 na página 11.

Comportamento	% eventos sensoriados
Só aleatório	23,2%
Misto	72,8%
Só gradiente	88,9%

Tabela 5.2: Chuva: Eventos sensoriados por pelo menos um agente.

Na Tabela 5.2, foi mostrado o valor global durante toda a execução da simulação. A seguir, é mostrado um gráfico da evolução desses valores com o decorrer do tempo.

Esse (e todos os demais gráficos do gênero presentes neste capítulo) dividiram o tempo simulado em 10 partes iguais e calcularam os valores da Tabela 5.2 para os eventos de cada uma das partes. O resultado está na Figura 5.2.

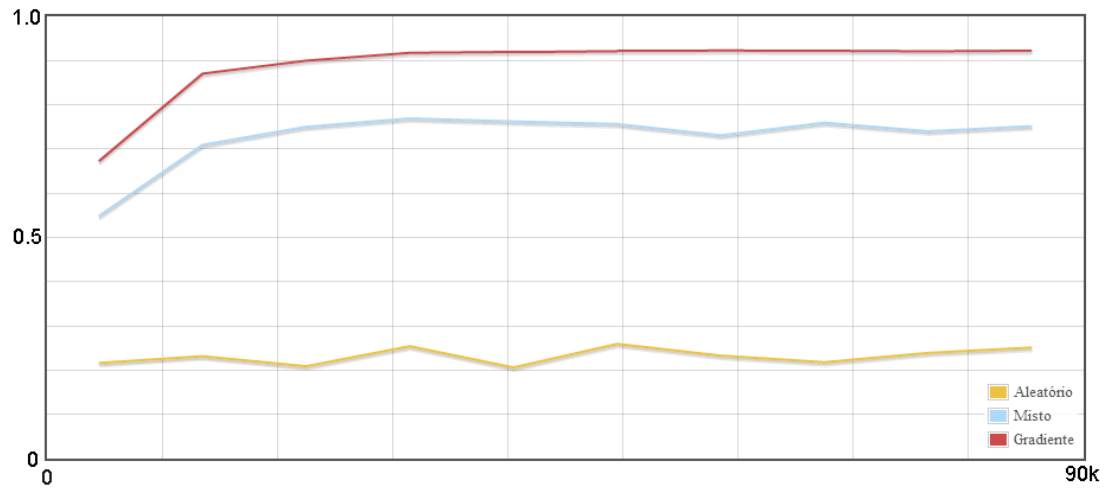


Figura 5.2: Evolução do número de eventos detectados com o tempo.

Parcela média de eventos que os agentes tomam conhecimento

Na Tabela 5.3, são mostrados os valores da parcela média de eventos que os agentes tomam conhecimento aleatório que tenha ocorrido. Mais detalhes sobre esta métrica estão na Seção 3.3.2 na página 12.

Comportamento	Parcela média de agentes
Só aleatório	18,6%
Misto	57,3%
Só gradiente	83,7%

Tabela 5.3: Chuva: Parcela média de eventos que os agentes tomam conhecimento.

Na Tabela 5.3, novamente foi mostrado o valor médio durante toda a simulação. A Figura 5.3 mostra a evolução desses valores com o decorrer do tempo.

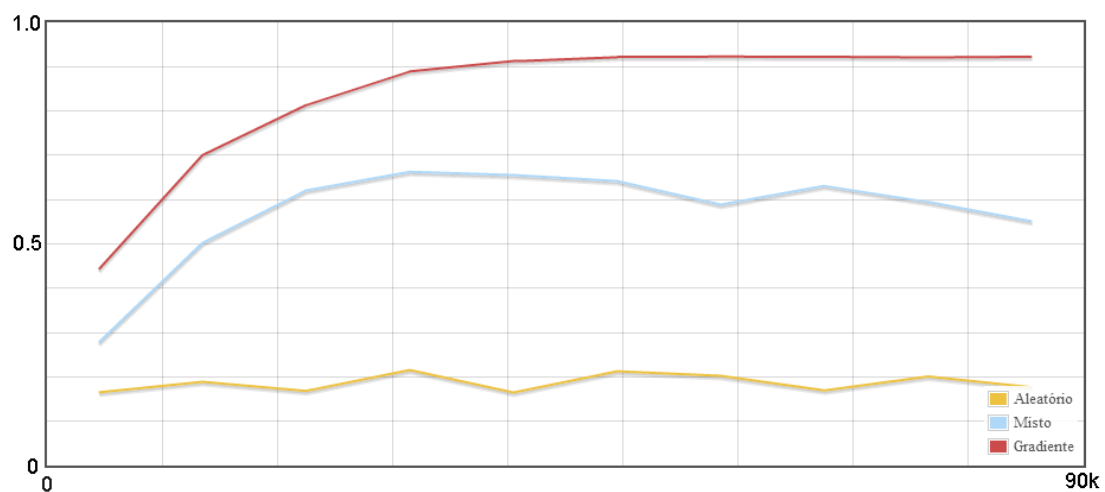


Figura 5.3: Evolução da parcela média de eventos que os agentes conhecem.

5.1.3 Resultados na internet

Demais resultados, todos os parâmetros escolhidos, os eventos utilizados, além dos resultados já mostrados, estão disponíveis para consulta através do endereço:

<http://www.cos.ufrj.br/~rodrigoroma/dissertacao/?experimento=chuva>

5.1.4 Comentários

Neste experimento, devido ao fato de que todos os eventos ocorrem em uma faixa contínua conforme a Figura 5.1, tem-se que, quanto maior o número de agentes no modo gradiente, mais agentes estarão sensoriando essa faixa vertical onde há eventos ocorrendo. Logo, a eficácia será maior, conforme comprovado pelos resultados.

No caso do comportamento misto, o número de eventos detectados foi um pouco menor que no caso do comportamento gradiente porém havia uma área maior coberta pelos sensores, apesar de que parte dessa cobertura foi desperdiçada dado que os eventos só ocorreram em uma única faixa.

5.2 Combate à poluição

O segundo experimento mostrado nesta dissertação consiste na simulação da detecção de partículas nocivas ao meio ambiente.

Cada partícula nociva que surge é considerada um evento. A partícula, após surgir, permanece visível por 100 minutos antes que desapareça.

5.2.1 Parâmetros

Eventos e duração

Neste experimento, durante 100k minutos (tempo simulado no experimento) ocorrem eventos de acordo com as densidades relativas mostradas na Figura 5.4.

Foram gerados um total 750k eventos para este experimento, cujo tempo de visibilidade é $T_{vis} = 100$ e cujo tempo de interesse é $T_{int} = 1000$. O tempo de visibilidade 100 define uma segunda classe de problemas onde o tempo de visibilidade é um valor próximo ao tempo de sensoriamento.

Considere as letras identificadoras A, B, C e D das sub-regiões mostradas na Figura 5.5. Para a sub-região A, foram gerados 200k eventos aleatórios tanto no tempo quanto na posição. Para a sub-região B, foram gerados 150k eventos. Como os eventos em A ocorrem durante todo o experimento e os em B, durante apenas 3/4 do tempo total simulado, A e B possuem a mesma densidade média.

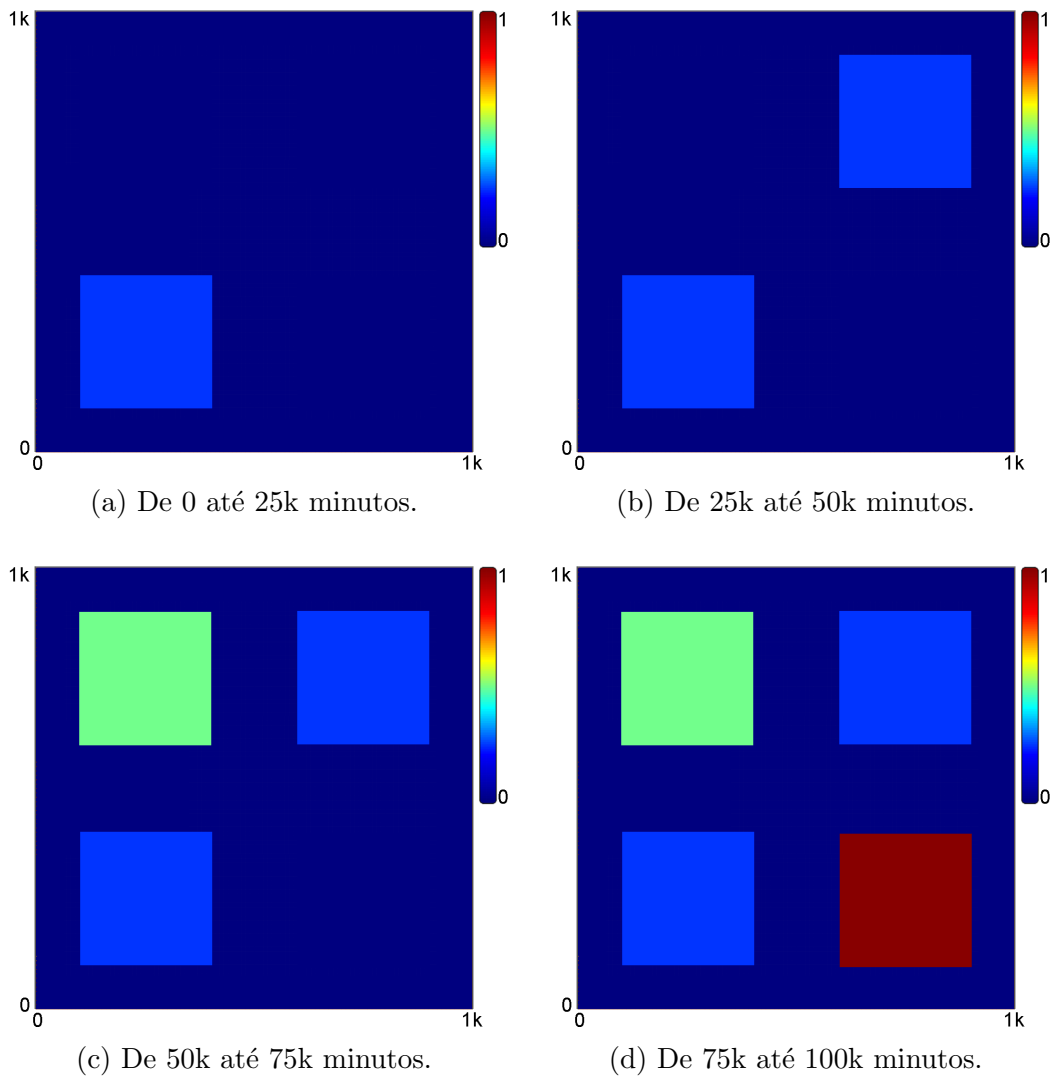


Figura 5.4: Densidade de eventos idealizada. Os eventos reais utilizados estão disponíveis na página do experimento.

Na sub-região C, há 200k eventos ocorrendo. Como os eventos em C ocorrem apenas durante 1/2 do tempo simulado (conforme explicado na Figura 5.4), tem-se que a densidade em C é o dobro da densidade de A e B.

Na sub-região D, também há 200k eventos, porém estes acontecem apenas no quarto final do experimento, de forma que a densidade desta sub-região é quatro vezes a densidade de A e B e o dobro da densidade de C e também a maior do experimento.

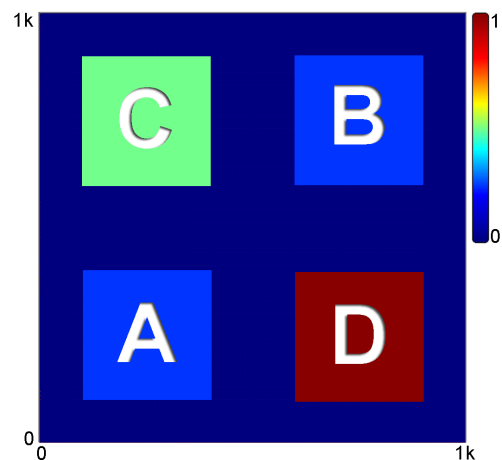


Figura 5.5: Identificadores.

Número e características dos agentes

Foram utilizados 50 agentes com um raio de sensoriamento de 100 metros cada.

Comportamento dos agentes

Novamente foram analisados 3 comportamentos distintos. Relembrando:

- **Comportamento aleatório:** os agentes só operam no modo aleatório.;
- **Comportamento misto:** após entrarem no modo gradiente, os agentes voltam ao modo aleatório com certa probabilidade e vice versa;
- **Comportamento gradiente:** após entrar no modo gradiente, o agente permanece neste modo enquanto for possível.

Em todos os 3 casos acima, os agentes sempre escolherão o tempo em que eles permanecerão imóveis através de uma distribuição normal $N(25, 1)$.

Tabela de parâmetros

Na Tabela 5.4, são apresentados todos os parâmetros utilizados neste experimento.

Parâmetro	Comportamento		
	Aleatório	Misto	Gradiente
$R_L \times R_A$	1000x1000	1000x1000	1000x1000
$D_L \times D_A$	100x100	100x100	100x100
T	100000	100000	100000
$N(m, \sigma)$	$N(25, 1)$	$N(25, 1)$	$N(25, 1)$
A_{minS}	9999	50	0
$A_{threshold}$	9999	0.01	0.001
$G_{threshold}$	9999	0.001	0.001
G_p	100%	5%	0
G_{maxG}	9999	0.05	0
G_{minV}	0	25	9999
S_r	10	10	10
N	50	50	50
S_a	25	25	25
S_g	25	25	25
R_s	100	100	100
R_c	200	200	200
P_c	90%	90%	90%
T_{vis}	100	100	100
T_{int}	1000	1000	1000

Tabela 5.4: Poluição: Tabela de parâmetros.

5.2.2 Resultados

Eventos sensoriados por pelo menos um agente

Na Tabela 5.5, são mostradas as parcelas de eventos (dentre os eventos que ocorreram durante toda a simulação) onde pelo menos um agente foi capaz de sensoriá-los.

Comportamento	% eventos sensoriados
Só aleatório	54,7%
Misto	83,0%
Só gradiente	45,2%

Tabela 5.5: Poluição: Eventos sensoriados por pelo menos um agente.

Na Tabela 5.5, foi mostrado o valor global durante toda a execução da simulação. Na Figura 5.6, o tempo é discretizado em 10 partes e é apresentado um gráfico da evolução desses valores com o decorrer do tempo.

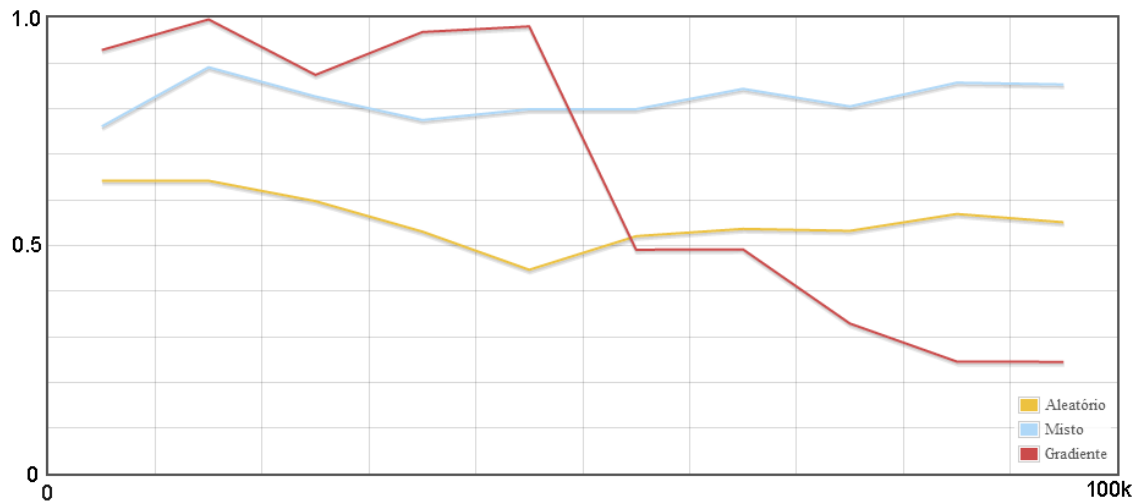


Figura 5.6: Evolução do número de eventos detectados com o tempo

Parcela média de eventos que os agentes tomam conhecimento

Na Tabela 5.6, são mostrados os valores da parcela média de eventos que os agentes tomam conhecimento aleatório que tenha ocorrido. A Figura 5.7 mostra a evolução desses valores com o decorrer do tempo.

Comportamento	Parcela média de agentes
Só aleatório	49,3%
Misto	60,9%
Só gradiente	24,5%

Tabela 5.6: Poluição: Parcela média de eventos que os agentes tomam conhecimento.

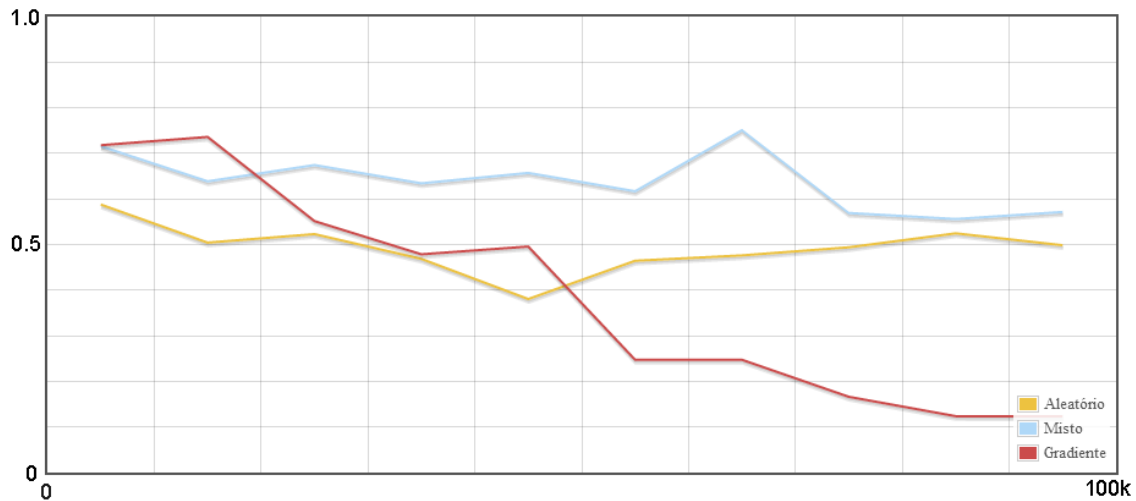


Figura 5.7: Evolução da parcela média de eventos que os agentes tomam conhecimento.

5.2.3 Resultados na internet

Demais resultados, todos os parâmetros escolhidos, os eventos utilizados, além dos resultados já mostrados, estão disponíveis para consulta através do endereço:

<http://www.cos.ufrj.br/~rodrigoroma/dissertacao/?experimento=poluicao>.

5.2.4 Comentários

Ao contrário do primeiro experimento, onde a qualidade do resultado aumentava conforme o número de agentes no modo gradiente também aumentava, neste segundo experimento houve uma drástica perda de performance quando os agentes foram configurados para seguirem o gradiente sempre que possível, chegando até mesmo a ter um desempenho inferior ao comportamento aleatório.

Observando, através da página do experimento, o gráfico interativo que contém a dinâmica da movimentação dos agentes, nota-se que, 50k minutos após o tempo inicial da simulação, todos os agentes já estão no modo gradiente (alguns estão na sub-região A e os outros, na B).

Ora, se todos os agentes estão em A ou B no modo gradiente e configurados para não saírem deste modo, após 50k minutos haverá eventos ocorrendo em C e após 75k minutos, em D. Porém, todos os agentes estarão “presos” em A ou B de forma que nenhum evento de C ou D será detectado. Como a maioria dos eventos ocorre justamente em C e D (mais precisamente, 53%), o desempenho fica muito baixo.

5.3 Chuva de granizo

Este experimento simula uma chuva de granizo. Ao atingir o solo, o granizo permanece visível por 1 hora até que este se derreta totalmente ficando indetectável.

5.3.1 Parâmetros

Eventos e duração

Neste experimento, durante 86400 segundos (ou seja, um dia inteiro) ocorrem eventos de acordo com as densidades relativas conforme a Figura 5.8 mostrada abaixo.

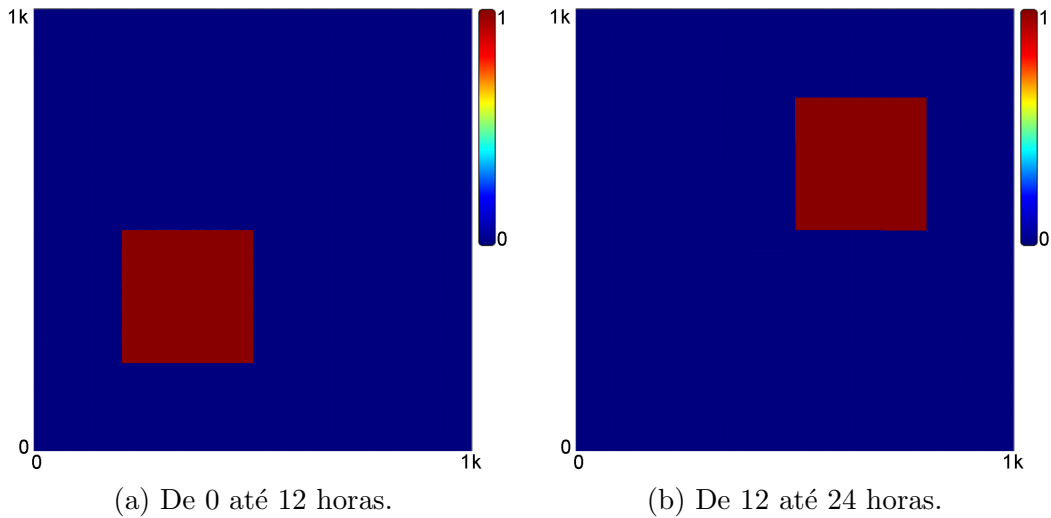


Figura 5.8: Densidade de eventos idealizada. Os eventos reais utilizados estão disponíveis na página do experimento.

Foram gerados um total 120k eventos para este experimento, cujo tempo de visibilidade é $T_{vis} = 3600$, ou seja, 1 hora, visto que a unidade de tempo utilizada foi o segundo. O tempo de interesse por eles é $T_{int} = 3600$, também 1 hora.

Aqui tem-se uma terceira classe de problemas, onde o tempo de visibilidade é muito maior que o tempo de sensoriamento (1 hora contra 1 minuto).

Número e características dos agentes

Nesta simulação, foram utilizados 15 agentes com raio de sensoriamento de 100m. Foi possível utilizar um número menor de agentes devido ao maior tempo de visibilidade (quanto maior o tempo de visibilidade, mais chances há de o evento ser detectado por um agente).

Comportamento dos agentes

Diferentemente dos demais experimentos mostrados até então, este não contempla um comportamento misto dos agentes devido ao baixo número de agentes que se tem

disponível (que resulta em uma baixa densidade de agentes na região), pois, caso contrário, haveria um número muito pequeno em cada um dos modos (aleatório e gradiente). Dessa forma, há apenas dois comportamentos:

- **Comportamento aleatório:** os agentes só operam no modo aleatório durante toda a simulação;
- **Comportamento gradiente:** após entrar no modo gradiente, o agente permanece neste modo enquanto for possível (mais precisamente, enquanto o módulo do gradiente de F for maior que 0,001).

Em todos os casos, os agentes sempre escolherão o tempo em que eles permanecerão imóveis através de uma distribuição normal $N(60, 1)$, onde 60 representa a média em segundos, ou seja, 1 minuto.

Tabela de parâmetros

Na Tabela 5.7, são apresentados todos os parâmetros utilizados neste experimento.

Parâmetro	Comportamento	
	Aleatório	Gradiente
$R_L \times R_A$	1000x1000	1000x1000
$D_L \times D_A$	100x100	100x100
T	86400	86400
$N(m, \sigma)$	$N(60, 1)$	$N(60, 1)$
A_{minS}	9999	0
$A_{threshold}$	9999	0.001
$G_{threshold}$	9999	0.001
G_p	100%	0
G_{maxG}	9999	0
G_{minV}	0	9999
S_r	10	10
N	15	15
S_a	25	25
S_g	25	25
R_s	100	100
R_c	200	200
P_c	90%	90%
T_{vis}	99999	99999
T_{int}	3600	3600

Tabela 5.7: Granizo: Tabela de parâmetros.

5.3.2 Resultados

Eventos sensoriados por pelo menos um agente

Na Tabela 5.8, são mostradas as parcelas de eventos (dentre os eventos que ocorreram durante toda a simulação) onde pelo menos um agente foi capaz de sensoriá-lo.

Comportamento	% eventos sensoriados
Só aleatório	61,1%
Só gradiente	95,5%

Tabela 5.8: Granizo: Eventos sensoriados por pelo menos um agente.

Na Tabela 5.8, foi mostrado o valor global durante toda a execução da simulação. Na Figura 5.9 a seguir, o tempo é discretizado em 10 partes e é apresentado um gráfico da evolução desses valores com o decorrer do tempo.

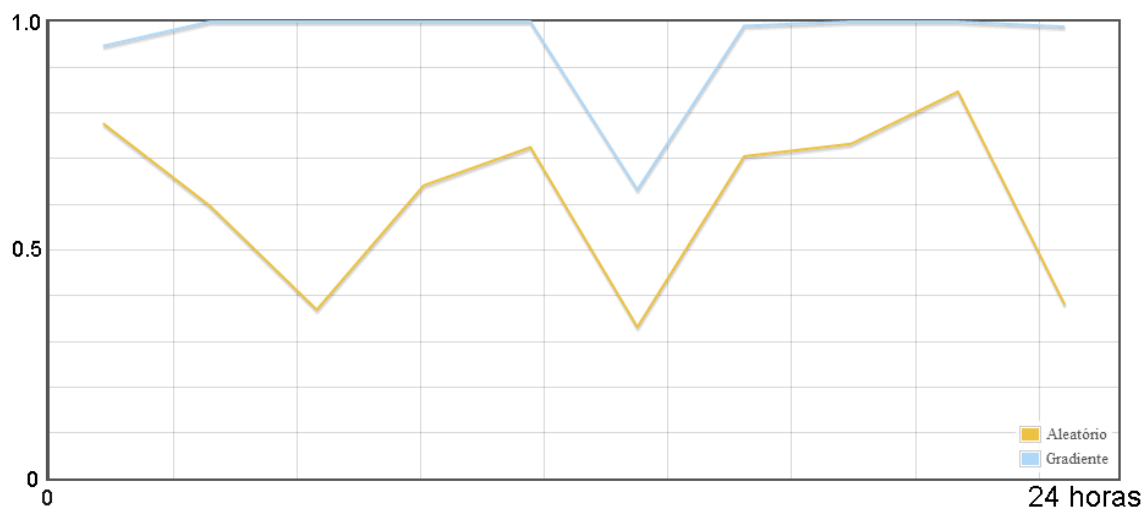


Figura 5.9: Evolução do número de eventos detectados com o tempo.

Parcela média de eventos que os agentes tomam conhecimento

Na Tabela 5.9, são mostrados os valores da parcela média de eventos que os agentes tomam conhecimento aleatório que tenha ocorrido. A Figura 5.10 mostra a evolução desses valores com o decorrer do tempo.

Comportamento	Parcela média de agentes
Só aleatório	20,2%
Só gradiente	75,7%

Tabela 5.9: Granizo: Parcela média de eventos que os agentes tomam conhecimento.

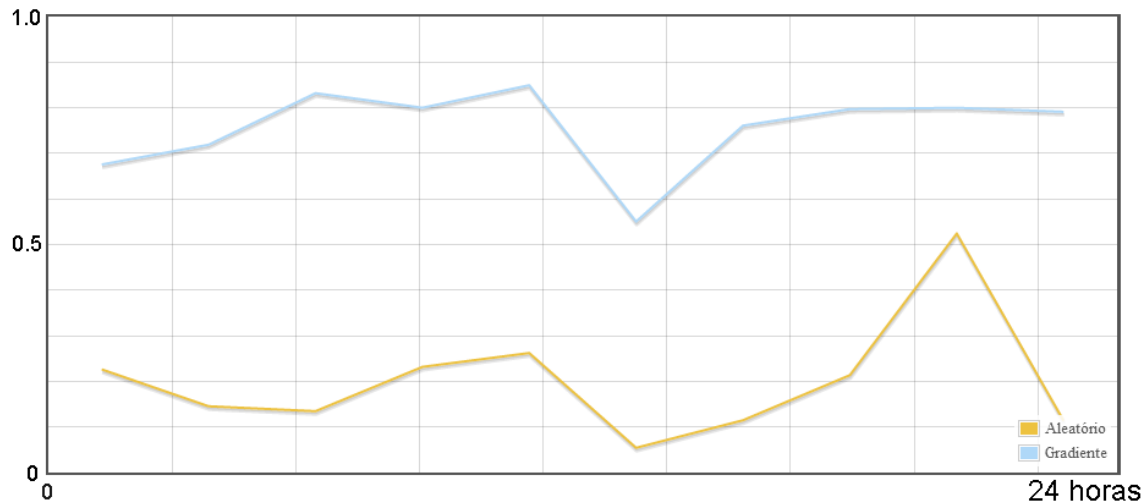


Figura 5.10: Evolução da parcela média de eventos que os agentes tomam conhecimento.

5.3.3 Resultados na internet

Demais resultados, todos os parâmetros escolhidos, os eventos utilizados, além dos resultados já mostrados, estão disponíveis para consulta através do endereço:

<http://www.cos.ufrj.br/~rodrigoroma/dissertacao/?experimento=granizo>.

5.3.4 Comentários

As principais diferenças deste experimento em relação aos demais é o grande tempo de visibilidade (3600 segundos) e o pequeno número de agentes (apenas 15).

Como o tempo de visibilidade é grande, tem-se que, para um determinado evento que tenha ocorrido porém não tenha sido sensoriado imediatamente por nenhum agente, ainda há uma grande chance de o mesmo ser sensoriado no futuro. Quanto maior o tempo de visibilidade, mais provável é que um agente qualquer passe próximo a este evento e o detecte.

Repare que, com apenas 15 agentes no modo aleatório, ainda assim 61,1% dos eventos foram detectados. Ao concentrar os agentes nos locais onde há eventos ocorrendo (comportamento gradiente), esse número aumentou para 95,5%.

Já a parcela média de eventos que os agentes tomam conhecimento aumentou de 20,2% para 75,7% ao aplicar o algoritmo definindo um comportamento gradiente em vez de aleatório. Esse ganho expressivo é justificado pelo fato de, ao concentrar os agentes onde há eventos, se torna muito mais provável que uma mensagem enviada por um agente chegue aos demais, visto que, dada a baixa densidade de agentes, as mensagens só conseguem uma rota para uma pequena fração de agentes.

Em ambos os gráficos, na linha que representa a evolução da métrica no modo

aleatório, há alguns picos que se justificam devido justamente à aleatoriedade do comportamento. Como o número de agentes é baixo, o fator aleatório de passar ou não um agente por determinado local de forma a detectar um evento é relevante, ao contrário de quando se tem diversos agentes, onde é mais fácil conseguir que haja aproximadamente o mesmo número de agentes sensoriando determinado local. Já na linha que representa o comportamento gradiente, vemos que há um decaimento no valor da métrica no momento em que o local de ocorrência dos eventos é alterado (ou seja, após 12h). Isto era o esperado visto que, alterando o local da ocorrência dos eventos, vai demorar um tempo até que os agentes encontrem esse novo local.

Capítulo 6

Conclusão

Este trabalho apresentou um algoritmo para sensoriamento de eventos dinâmicos através de agentes autônomos móveis, cuja motivação para abordarmos este assunto surgiu do crescente aumento do interesse por esta área.

O algoritmo apresentado foi concebido com o objetivo de poder ser implementado nos mais diversos cenários reais possíveis, ou seja, procuramos não partir de premissas que restrinjam em muito a implementação prática do algoritmo como, por exemplo, que o grafo de comunicação é completo (todo agente comunica-se diretamente com os demais) ou que todos os agentes já conhecem previamente a densidade e que esta é imutável, ambas as premissas são utilizadas em [2].

O modelo de comunicação adotado é simples, não havendo confirmação de recebimento das mensagens, e um dos motivos para isso foi a economia de energia que, apesar desta não ter sido explicitamente considerada, esteve presente quando a viabilidade da implementação foi questionada.

6.1 Contribuições

A seguir, são apresentadas as principais contribuições deste trabalho.

Densidade de eventos dinâmica

O algoritmo proposto foi pensado com o objetivo de satisfazer o maior número possível de tipos de eventos. Pode-se dizer que o algoritmo apresentado é capaz de lidar com as seguintes situações:

- A densidade pode seguir qualquer função;
- A densidade pode variar com o tempo.

Dessa forma, não há restrições quanto aos cenários de eventos aos quais é possível implementar este algoritmo.

Modo gradiente e modo aleatório

Para que o algoritmo possa se adequar a qualquer densidade de eventos, os agentes não devem apenas seguir o gradiente de F , pois, caso contrário, eles só conseguiriam atingir máximos locais, deixando, eventualmente, diversos outros pontos de máximo descobertos. O algoritmo apresentado nesta dissertação contribui com uma proposta que faz com que o agente ande na direção do gradiente, porém também, eventualmente, pare de seguir o gradiente e ande aleatoriamente com o objetivo de encontrar outros pontos de máximo. Com isso, foi introduzido o conceito de **modo aleatório** e **modo gradiente**.

Como condição para trocar do modo aleatório para o modo gradiente e vice versa, esta proposta permite que diversos parâmetros sejam ajustados de forma a melhorar a eficácia dos agentes fazendo com que os mesmos, por exemplo, tendam a permanecer mais no modo aleatório ou mais no modo gradiente, onde tal escolha de comportamento vai depender do conhecimento prévio que se tem a respeito dos eventos. Em um caso extremo, pode-se, inclusive, configurar o agente para, ao entrar no modo gradiente, nunca sair do mesmo.

Tempo de visibilidade

Outra contribuição deste trabalho é permitir que os eventos possam ser modelados incluindo o tempo de visibilidade dos mesmos. Ou seja, é possível modelar eventos que ocorrem instantaneamente bem como modelar eventos que, após acontecerem, permanecem visíveis por algum tempo até que finalmente não seja mais possível detectá-los.

Atualização da densidade

Este trabalho também contribuiu ao tornar possível configurar o tempo de interesse nos eventos, ou seja, ao modelar como é calculada a densidade de eventos ϕ baseando-se na amostra que se tem dos eventos que foram detectados.

Alguns trabalhos relacionados haviam sugerido que a densidade fosse sendo atualizada conforme novos eventos fossem detectados mas não chegaram a modelar como esta atualização da densidade seria feita. Por exemplo, em [9] é mencionado que os agentes iniciam a atuação contendo uma mesma visão da densidade que foi pré-configurada em todos eles e que, com o tempo, os agentes vão atualizando a densidade baseados em suas próprias detecções e nas detecções de seus vizinhos cuja informação foi recebida pela rede de comunicação. Em outros trabalhos é considerado, inclusive, que a densidade deve ser estática.

Aprimoramentos

Contribuímos também nas seguintes questões:

- Os agentes mais próximos à “borda” do local onde estão ocorrendo eventos deverão ser mais suscetíveis a deixarem o modo gradiente do que os agentes próximos ao “centro” de tal local;
- Ao deixar o modo gradiente, o agente deverá executar S_r passos na direção que o afasta dos eventos, de forma a garantirmos que o mesmo não permanecerá próximo aos eventos executando o mesmo sensoriamento de antes ao invés de explorar novas localidades.

Ao introduzir essas duas ideias no algoritmo proposto, este foi aprimorado, pois os agentes obtiveram um comportamento ainda mais próximo do comportamento que se desejava que consiste em os agentes que estão no modo aleatório estarem explorando uma localidade onde não há agentes no modo gradiente pois, caso contrário, o agente no modo aleatório estaria desempenhando o mesmo papel dos agentes que estão no modo gradiente e próximos a ele e, neste caso, para maximizar, seria melhor que ele também estivesse no modo gradiente.

6.2 Limitações e trabalhos futuros

Este trabalho possui algumas limitações que serão analisadas a seguir. Para cada uma dessas limitações, será proposta uma modificação a ser realizada em um trabalho futuro.

Modelagem da comunicação

Neste trabalho, a comunicação entre os agentes está modelada através da definição de um raio de comunicação, ou seja, distância máxima que permite que dois agentes se comuniquem, e também uma probabilidade de sucesso no recebimento da mensagem enviada (que é aplicada a cada agente dentro desse raio de comunicação).

Este modelo foi adotado devido à sua simplicidade, porém sabe-se que, quanto mais próximos os agentes estiverem, maior será o sucesso no recebimento de pacotes enviados pelo meio *wireless*.

Como proposta, a modelagem pode ser aprimorada de forma a contemplar, também, o decaimento da probabilidade de haver sucesso no recebimento da mensagem conforme a distância entre os agentes aumente.

Tempo de visibilidade não fixo

Neste trabalho, é definido por quanto tempo um evento será visível após a sua ocorrência, ou seja, por quanto tempo ainda será possível um agente detectá-lo. Este tempo é um valor fixo.

Dependendo do evento que está sendo modelado, pode ser que os eventos ocorram com diferentes intensidades. Por exemplo, um raio que deixa sua marca no solo pode deixar uma marca mais forte ou mais fraca de acordo com a sua intensidade. Quanto mais forte a marca, por mais tempo será possível detectar que um raio ocorreu naquele ponto.

Como este trabalho está limitado a modelar eventos de forma que todos tenham o mesmo tempo de visibilidade, uma proposta de trabalho futuro é implementar o tempo de visibilidade como sendo uma variável aleatória ou que, para cada evento especificado, defina-se, além de sua posição, por quanto tempo o mesmo ainda será visível.

Marca deixada pelo evento ficando mais fraca

Após um evento ocorrer, se o mesmo continuar visível por algum tempo, é sinal de que ele deixou de alguma forma uma marca de que ele ocorreu.

A modelagem dessa situação neste trabalho considera que essa marca deixada pelo evento permanecerá visível por exatamente T_{vis} unidades de tempo, porém a probabilidade de detectá-lo será a mesma ao longo de todo este intervalo de tempo.

Em alguns casos, a marca deixada pelo evento vai enfraquecendo com o tempo. Após uma queimada na floresta, por exemplo, as marcas deixadas pelo fogo vão sumindo dando novamente lugar à vida. Entretanto, esse comportamento não está sendo modelado neste trabalho.

Como proposta, é possível, também, modelar a visibilidade de um evento não só como “visível” ou “não visível” (como é o caso atual) mas também assumir que a capacidade dos agentes de detectá-lo vai decaindo com o passar do tempo após sua ocorrência, ou seja, a marca deixada vai ficando cada vez mais fraca até que suma totalmente.

Tempo de deslocamento não desprezível

Apesar de o gasto de energia (com comunicação, sensoriamento e deslocamento) não ser explicitamente considerado neste trabalho, ainda assim o algoritmo foi pensado com o objetivo de ser implementado nos mais diversos cenários reais possíveis e, para isso, é necessário lembrar que a energia disponível em um agente é finita. Visando minimizar o consumo de energia com o deslocamento, foi considerado que o tempo de sensoriamento é muito grande em relação ao tempo de deslocamento e, dessa

forma, foi considerado que o tempo gasto com o deslocamento é desprezível, ou seja, foi considerado como sendo sempre nulo.

Entretanto, em alguns cenários, isso pode não ser razoável. Por motivos diversos, pode ser interessante que o tempo de deslocamento não seja desprezível quando comparado ao tempo de sensoriamento. Por este motivo, uma proposta para aprimorar este trabalho é acrescentar um parâmetro para especificar a velocidade com a qual o agente se desloca. Pode-se, também, especificar se o agente é capaz de se mover e sensoriar simultaneamente ou se ele só consegue sensoriar quando está parado.

Parâmetros adaptativos

Apesar de o algoritmo apresentado ser bastante robusto e capaz de lidar com diversas situações, o trabalho de ajustar os parâmetros manualmente para que os agentes executem um sensoriamento com a qualidade (valor desejável das métricas) esperada ainda é bastante desgastante pois consiste em testar as diversas variações dos parâmetros até descobrir com qual valor o resultado é melhor. Como há diversos parâmetros a serem ajustados, isso pode ser bastante trabalhoso.

A última proposta de trabalho futuro desta dissertação consiste em tornar o algoritmo aqui presente auto-ajustável. Alguns parâmetros serão necessariamente fixos pois dependem das características dos agentes utilizados (raio de sensoriamento, por exemplo). Entretanto, os parâmetros relacionados ao comportamento dos agentes podem ser ajustados dinamicamente com o objetivo de maximizar a detecção dos eventos sem que seja necessária uma configuração manual.

Referências Bibliográficas

- [1] MARTINEZ, S., CORTES, J., BULLO, F. “Motion coordination with distributed information”, *IEEE Control Systems Magazine*, v. 27, n. 4, pp. 75–88, ago. 2007.
- [2] GUSRIALDI, A., DIRZA, R., HIRCHE, S. “Information-driven distributed coverage algorithms for mobile sensor networks”. In: *2011 International Conference on Networking, Sensing and Control, ICNSC 2011*, pp. 242–247, 2011.
- [3] NENE, M. J., DEODHAR, R. S., PATNAIK, L. M. “UREA: Uncovered Region Exploration Algorithm for reorganization of mobile sensor nodes to maximize coverage”. In: *DCOSS '10 - International Conference on Distributed Computing in Sensor Systems, Adjunct Workshop Proceedings: IWSN, MobiSensors, Poster and Demo Sessions*, 2010.
- [4] JI, M., EGERSTED, M. “A Graph-Theoretic Characterization of Controllability for Multi-agent Systems”. In: *American Control Conference, 2007. ACC '07*, pp. 4588 –4593, july 2007. doi: 10.1109/ACC.2007.4283010.
- [5] DIRZA, R., GUSRIALDI, A. “Performance-guaranteed distributed coverage control for robotic visual sensor network with limited energy storage”. In: *Proceedings of 2011 2nd International Conference on Instrumentation Control and Automation, ICA 2011*, pp. 334–339, 2011.
- [6] ZOU, Y., CHAKRABARTY, K. “Sensor deployment and target localization based on virtual forces”. In: *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, v. 2, pp. 1293 – 1303 vol.2, march-3 april 2003. doi: 10.1109/INFCOM.2003.1208965.
- [7] A. HOWARD, M. J. M., SUKHATME, G. S. “Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem”. In: *Proc. 6th Int. Symposium on Distributed Autonomous Robotics Systems*, pp. 299–308, Fukuoka, Japan, jun. 2002.

- [8] PODURI, S., SUKHATME, G. S. “Constrained coverage for mobile sensor networks”. In: *Proceedings - IEEE International Conference on Robotics and Automation*, v. 2004, pp. 165–171, 2004.
- [9] LI, W., CASSANDRAS, C. G. “Distributed Cooperative coverage Control of Sensor Networks”. In: *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pp. 2542 – 2547, dec. 2005.
- [10] ZHONG, M., CASSANDRAS, C. G. “Distributed coverage control and data collection with mobile sensor networks”, *IEEE Transactions on Automatic Control*, v. 56, n. 10, pp. 2445–2455, 2011.
- [11] CASSANDRAS, C. G., ZHONG, M. “Distributed coverage control in sensor network environments with polygonal obstacles”. In: *IFAC Proceedings Volumes (IFAC-PapersOnline)*, v. 17, 2008.
- [12] PERES, Y., SINCLAIR, A., SOUSI, P., et al. “Mobile geometric graphs: detection, coverage and percolation”, *Probability Theory and Related Fields*, pp. 1–33, 2012.
- [13] KESIDIS, G., KONSTANTOPOULOS, T., PHOHA, S. “Surveillance coverage of sensor networks under a random mobility strategy”. In: *Proceedings of IEEE Sensors*, v. 2, pp. 961–965, 2003.
- [14] LI, W., CASSANDRAS, C. G. “Distributed cooperative coverage control of sensor networks”. In: *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, CDC-ECC '05*, v. 2005, pp. 2542–2547, 2005.
- [15] PENROSE, M. *Random Geometric Graphs*. Oxford University Press, 2003.