



**Unioeste - Universidade Estadual do Oeste do Paraná**  
**CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS**  
Colegiado de Ciência da Computação  
*Curso de Bacharelado em Ciência da Computação*

**Um Modelo Multiagente *Bitstring* em CUDA para Simular a Propagação de Hipotéticas Doenças Baseadas em Modelagem Compartmental Tipo SEIRS**

*Wesley Luciano Kaizer*

**CASCABEL**  
**2016**

**WESLEY LUCIANO KAIZER**

**UM MODELO MULTIAGENTE *BITSTRING* EM CUDA PARA  
SIMULAR A PROPAGAÇÃO DE HIPOTÉTICAS DOENÇAS BASEADAS  
EM MODELAGEM COMPARTIMENTAL TIPO SEIRS**

Monografia apresentada como requisito parcial  
para obtenção do grau de Bacharel em Ciência da  
Computação, do Centro de Ciências Exatas e Tec-  
nológicas da Universidade Estadual do Oeste do  
Paraná - Campus de Cascavel

Orientador: Prof. Dr. Rogério Luís Rizzi

CASCABEL  
2016

**WESLEY LUCIANO KAIZER**

**UM MODELO MULTIAGENTE *BITSTRING* EM CUDA PARA  
SIMULAR A PROPAGAÇÃO DE HIPOTÉTICAS DOENÇAS BASEADAS  
EM MODELAGEM COMPARTIMENTAL TIPO SEIRS**

Monografia apresentada como requisito parcial para obtenção do Título de Bacharel em  
Ciência da Computação, pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel,  
aprovada pela Comissão formada pelos professores:

---

Prof. Dr. Rogério Luís Rizzi (Orientador)  
Colegiado de Matemática, UNIOESTE

---

Profa. Dra. Claudia Brandelero Rizzi  
Colegiado de Ciência da Computação,  
UNIOESTE

---

Prof. Dr. Guilherme Galante  
Colegiado de Ciência da Computação,  
UNIOESTE

Cascavel, 9 de julho de 2016

## **DEDICATÓRIA**

## **AGRADECIMENTOS**

# Lista de Figuras

2.1	Fluxo compartimental no modelo SEIRS . . . . .	10
3.1	Representação da tira do agente $\chi$ . . . . .	24

# Lista de Tabelas

2.1	Tabela verdade para a operação unária de negação . . . . .	18
2.2	Tabela verdade para a operação binária de conjunção . . . . .	18
2.3	Tabela verdade para a operação binária de disjunção inclusiva . . . . .	19
2.4	Tabela verdade para a operação binária de disjunção exclusiva . . . . .	19
3.1	Tabela de tamanhos . . . . .	27
3.2	Tabela de complementares . . . . .	27
3.3	Tabela de posteriores . . . . .	27
3.4	Tabela de anteriores . . . . .	27

# Lista de Abreviaturas e Siglas

SIS	Modelo Suscetível-Infectado-Suscetível
SIR	Modelo Suscetível-Infectado-Recuperado
SEIR	Modelo Suscetível-Exposto-Infectado-Recuperado
SEIRS	Modelo Suscetível-Exposto-Infectado-Recuperado-Suscetível
API	<i>Application Programming Interface</i>
CUDA	<i>Compute Unified Device Architecture</i>
CPU	<i>Central Processing Unit</i>
GPU	<i>Graphics Processing Unit</i>
GPGPU	<i>General Purpose Graphics Processing Unit</i>
EDO	Equações Diferenciais Ordinárias
PVI	Problema de Valor Inicial
PVIC	Problema de Valor Inicial e de Contorno
AC	Autômato Celular
LGCA	<i>Lattice Gas Cellular Automata</i>



# Lista de Símbolos

$\alpha$	Taxa de infecção
$\gamma$	Período de exposição
$\alpha$	Período de infectância
$\delta$	Período de recuperação
$\chi$	Agente Computacional
$\lambda$	Operador de evolução
$\mu$	Operador de transição espacial
$\sigma$	Operador de transição temporal

# Sumário

<b>Lista de Figuras</b>	<b>vi</b>
<b>Lista de Tabelas</b>	<b>vii</b>
<b>Lista de Abreviaturas e Siglas</b>	<b>viii</b>
<b>Lista de Símbolos</b>	<b>ix</b>
<b>Sumário</b>	<b>x</b>
<b>Resumo</b>	<b>xii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Objetivos . . . . .	2
1.2 Motivação e Justificativas . . . . .	3
1.3 Organização do Trabalho . . . . .	3
<b>2 Fundamentos</b>	<b>5</b>
2.1 Epidemiologia Matemática e Computacional . . . . .	5
2.2 Modelos compartimentais: Tipos e Classificações . . . . .	8
2.3 Agentes Inteligentes . . . . .	13
2.4 Metodologia de <i>Bitstring</i> . . . . .	17
<b>3 Metodologias Computacionais</b>	<b>21</b>
3.1 Introdução . . . . .	21
3.2 Modelagem em Operadores aos Agentes . . . . .	21
3.3 Modelagem em <i>Bitstring</i> aos Agentes . . . . .	23
3.4 Estruturas de Dados, Linguagens e Estratégias de Implementação . . . . .	28
3.5 CUDA e OpenMP . . . . .	28
3.6 SIMULA . . . . .	28

<b>4</b>	<b>Soluções</b>	<b>29</b>
4.1	Introdução . . . . .	29
4.2	Implementação Padrão com CUDA e OpenMP . . . . .	29
4.3	Implementação <i>Bitstring</i> com CUDA e OpenMP . . . . .	29
4.4	Discussões Qualitativas, Quantitativas, Eficiência e Acurácia . . . . .	29
<b>5</b>	<b>Resultados e Discussões</b>	<b>30</b>
5.1	Introdução . . . . .	30
5.2	Cases: Discutir as Simulações no Espaço Geográfico Escolhido. . . . .	30
	<b>Referências Bibliográficas</b>	<b>31</b>

# Resumo

A aplicação de modelagem compartimental na epidemiologia é amplamente estudada, como pode-se observar na extensa literatura disponível. A simulação de dinâmicas epidemiológicas é de particular interesse no estudo, prevenção e controle de doenças transmissíveis. Com base nestas premissas, este trabalho busca abordar o problema de simulação de hipotéticas doenças baseadas em modelagem compartimental SEIRS, por meio de sistema multiagente. Para obter uma adequada formulação à simulação em GPGPU na plataforma CUDA, ao modelo é empregada a metodologia de *bitstring*. Ao ambiente computacional são utilizadas técnicas e software especificamente desenvolvido para manipular dados georreferenciados à especificação e composição de um *lattice* apropriado à simulação computacional. Como resultado pretende-se apresentar uma aplicação funcional e adequada para simular eventos epidemiológicos em uma região da cidade de Cascavel/PR. A avaliação do modelo proposto é executada através da realização de experimentos numérico-computacionais, onde os resultados são comparados aos obtidos na literatura, sobretudo nos aspectos quantitativos às implementações realizadas.

**Palavras-chave:** Epidemiologia computacional, Sistema multiagente, Modelos compartimentais, Modelagem *bitstring*, Plataforma Computacional CUDA.

# Capítulo 1

## Introdução

Epidemiologia pode ser definida como o estudo da frequência, da distribuição e dos estados ou eventos relacionados ao espalhamento de doenças transmissíveis e ocorrências de doenças não transmissíveis em populações específicas, e a aplicação dos resultados desses estudos na prevenção e controle dos problemas decorrentes e relacionados com a saúde pública. Modelos computacionais baseados em indivíduos vêm sendo empregados na epidemiologia para estudar a propagação e a transmissão de doenças, que são processos centrais na dinâmica de doenças infecto-contagiosas. O uso destes modelos viabiliza a modelagem de fenômenos de natureza probabilística e da heterogeneidade nas relações entre os indivíduos e o meio, conferindo mais realidade ao modelo estudado. Modelos compartimentais podem ser utilizados à definição de modelos mais complexos, tendo como base a subdivisão da população em categorias, em que os indivíduos fluem entre elas de acordo com determinadas taxas e cenários, podendo respeitar as características particulares de uma doença de interesse.

À implementação de modelos baseados em indivíduos em uma linguagem computacional pode ser relevante utilizar abordagens mais eficientes à codificação do sistema e definir um *lattice* apropriado à execução dos experimentos computacionais. Dependendo da dimensão do *lattice*, da quantidade de indivíduos e da complexidade das dinâmicas modeladas, é desejável otimizar o tempo de execução dos experimentos, utilizando os recursos computacionais de processamento e armazenamento disponíveis da forma mais eficiente possível.

## 1.1 Objetivos

O objetivo principal deste trabalho é propor, desenvolver, implementar e avaliar um modelo epidemiológico computacional multiagente, com formulação em *bitstring*, para simular computacionalmente a propagação de doenças que possam ser modeladas por modelos compartimentais tipo Suscetível-Exposto-Infectado-Recuperado-Suscetível, SEIRS, utilizando como *lattice* um mapeamento georreferenciado de uma região da cidade de Cascavel/PR. A solução computacional do modelo proposto contempla uma implementação utilizando *Compute Unified Device Architecture*, CUDA, para obter máxima eficiência computacional por meio de paralelismo de dados. Para alcançar este objetivo é necessária a realização de objetivos mais específicos que contemplam:

1. Revisão bibliográfica nas temáticas pertinentes ao trabalho, incluindo temas como epidemiologia matemática e computacional, modelagem compartimental, agentes inteligentes e sistemas multiagente, formulação *bitstring*, estruturas de dados e plataforma computacional paralela CUDA.
2. Desenvolvimento e implementação de um modelo multiagente em *bitstring*, baseado em formulação compartimental, tendo como *lattice* um mapeamento georreferenciado de uma quadra da cidade de Cascavel.
3. Paralelização em nível de dados do sistema multiagente em *bitstring*, utilizando a plataforma computacional paralela CUDA.
4. O emprego e o aperfeiçoamento de uma ferramenta computacional para viabilizar e otimizar as fases de pré-processamento, processamento e pós-processamento da simulação, como as etapas de configuração e visualização dos resultados obtidos, utilizando o *lattice* gerado por tal software.
5. Realização de experimentos numérico-computacionais, visando avaliar a acurácia da solução implementada, bem como sua eficiência computacional, e demais aspectos computacionais.

## 1.2 Motivação e Justificativas

Em conformidade com os objetivos estabelecidos, este trabalho é motivado e se justifica pelos seguintes itens:

- O estudo e aplicação de modelos compartimentais em epidemiologia computacional é relevante, pois mostram-se adequados e flexíveis à modelagem de hipotéticas doenças, bem como são amplamente utilizados, como apresentado na literatura técnica.
- O uso de agentes computacionais em simulações permite a modelagem mais realística dos fenômenos epidemiológicos de interesse, pois os indivíduos têm rica e emergente dinâmica, independentemente uns dos outros, além de viabilizar o emprego de metodologia à sua paralelização.
- A abordagem em multiagente com *bitstring* é relativamente nova e relevante, viabilizando a modelagem de agentes computacionais de forma concisa e eficiente, otimizando o consumo de memória, simplificando os processos de captura e configuração de atributos dos indivíduos e facilitando a implementação na plataforma CUDA por reduzir significativamente a quantidade de dados nas transferências entre CPU e GPU, além de permitir o uso de estruturas de dados simplificadas.
- O uso da plataforma CUDA é atrativo por possibilitar a paralelização massiva do sistema implementado, acarretando ganhos de desempenho desejáveis nos experimentos computacionais que serão realizados.
- Por fim, o trabalho realizado é relevante à formação técnica do futuro profissional em Ciência da Computação.

## 1.3 Organização do Trabalho

Este trabalho apresenta a seguinte organização: no Capítulo 2 é apresentada e discutida as fundamentações teóricas utilizadas para a realização do trabalho de conclusão de curso, envolvendo temáticas como a epidemiologia matemática e computacional, modelagem compartimental, agentes inteligentes e sistemas multiagente e modelagem *bitstring*. No Capítulo 3 são

apresentados os métodos utilizados à modelagem, implementação e testes do sistema multiagente, como estruturas de dados, linguagens, APIs e demais softwares de apoio. No Capítulo 4 são discutidas as implementações realizadas através do uso das técnicas apresentadas nos capítulos 2 e 3, fazendo-se comparações e discutindo demais itens pertinentes. Por fim, o Capítulo 5 discute os resultados obtidos por meio da execução de testes e apresenta conclusões sobre as implementações realizadas.



# Capítulo 2

## Fundamentos

### 2.1 Epidemiologia Matemática e Computacional

Acredita-se que os primeiros estudos epidemiológicos foram realizados na Grécia Antiga pelo sacerdote Hipócrates (460-377 a.C.). Hipócrates investigou diversas epidemias e suas distribuições ambientais e, após sua morte, seus discípulos deram continuidade aos seus estudos, tomando diferentes frentes de trabalho. Em Roma, os trabalhos epidemiológicos foram iniciados por médicos inspirados em Galeno (201-130 a.C.), um famoso médico grego. Os estudos romanos produziram resultados importantes na área de epidemiologia, destacando-se a introdução dos censos periódicos e o registro compulsório de nascimentos e óbitos. Em 1950 aconteceu na Inglaterra a *London Epidemiological Society* organizada pela sociedade *Royal Medical Society*, onde participaram diversos profissionais proeminentes da área epidemiológica. Entre eles estava John Snow (1813-1858), considerado por muitos o pai da epidemiologia, que conduziu estudos pioneiros sobre a mortalidade por infecção pós-cirúrgica nos hospitais militares na Guerra da Criméia, sobre mecanismos de transmissão hídrica e realizou a descoberta do agente microbiano da cólera morbo. [Medronho et al. 2008].

Ocorreu ainda no século XIX um grande desenvolvimento na área de epidemiologia, destacando-se figuras como Major Greenwood (1888-1949), que introduziu o raciocínio estatístico na pesquisa epidemiológica, contribuindo para a epidemiologia experimental, John Ryle (1889-1950), que propôs a sistematização da História Natural das Doenças e Jerome Cornfield (1921-1979), que desenvolveu os estimadores de risco relativo e introduziu técnicas de regressão logística na epidemiologia. Nas décadas de 1960 e 1970 ocorreram grandes transformações na epidemiologia, com a introdução dos computadores e desenvolvimento de técnicas

de coleta e análise de dados epidemiológicos. Atualmente a epidemiologia faz uso de abordagens metodológicas modernas que contribuem para a reavaliação de suas bases de estudo, abrindo possibilidades para a investigação de grandes processos em grupos populacionais. O caráter inovador é bastante incentivado dentro da área, seja pela incorporação de novos paradigmas e modelos teóricos ou pela ampliação de objetos de estudos e uso de novas tecnologias. [Medronho et al. 2008].

No Brasil durante o século XIX aconteceram diversos movimentos importantes, com notável contribuição de pesquisadores brasileiros no desenvolvimento da epidemiologia no país. Dentre os mais importantes destacam-se os esforços de Oswaldo Cruz e Carlos Chagas. Oswaldo Cruz trabalhou no saneamento da cidade do Rio de Janeiro, na época capital do país, e no combate de epidemias como a febre amarela, peste bubônica e varíola, que contaram com forte auxílio militar. Carlos Chagas conduziu estudos sobre a malária, conseguindo controlar um surto epidêmico que ocorria na cidade de Itatinga, interior de São Paulo, sendo que seu trabalho é considerado, ainda hoje, referência mundial no combate à doença. Carlos Chagas também descobriu o protozoário causador da tripanossomíase americana, que posteriormente ficou mundialmente conhecida como a *Doença de Chagas*. [Medronho et al. 2008].

Em [Medronho et al. 2008] define-se a epidemiologia como o "*estudo da distribuição e dos determinantes dos eventos ou padrões de saúde em populações definidas, e a aplicação deste estudo para controlar problemas de saúde*", sendo que sua principal diferença à medicina clínica é no uso de populações em seus estudos. Esse diferencial justifica-se através do objetivo final da epidemiologia, que é a de melhorar o perfil de saúde das populações, e não somente de seus indivíduos em particular e que, inferências sobre a relação de determinados fatores e a ocorrência de doenças, somente podem ser feitas através do estudo de populações. [Medronho et al. 2008].

Segundo [Yang 2001] a epidemiologia matemática é fundamentada em hipóteses matemáticas que podem descrever aspectos de fenômenos biológicos nas interações entre hospedeiros e parasitas, utilizando-se de conhecimentos biológicos sobre o vírus e também da interação desse vírus com o hospedeiro ou com o meio ambiente. A partir destes conhecimentos desenvolvem-se modelos matemáticos que estão em constante evolução, de acordo com os avanços nos campos das ciências médicas e biológicas. A epidemiologia matemática tem então por finalidade descrever fenômenos observáveis e estudar mecanismos de intervenção externa

ao sistema de interação hospedeiro-parasita. Assim, a epidemiologia matemática agrega conhecimentos epidemiológicos e matemáticos no estudo de epidemias. Os métodos científicos aplicados à epidemiologia matemática permitem estudar tanto as situações de equilíbrio quanto as epidêmicas de uma doença ao longo do tempo. Definiu-se na epidemiologia matemática importantes conceitos como a teoria do valor limiar e o princípio da ação das massas. [Yang 2001], [Daley e Gani 1999].

- **Teoria do valor limiar:** Afirma que a introdução de uma baixa quantidade de indivíduos infectados em uma grande população de indivíduos suscetíveis não resultará em uma epidemia se o quantidade de suscetíveis não é maior que um determinado valor limiar crítico.
- **Princípio da ação das massas:** Estabelece que a propagação de epidemias depende da forma de interação ou contato entre os indivíduos suscetíveis e infectados na população, sendo o sucesso da propagação da doença dependente da mistura homogênea das classes de suscetíveis e infectados de indivíduos.

Em [Daley e Gani 1999] são discutidos, entre outros temas, os modelos matemáticos básicos desenvolvidos e estudados na epidemiologia matemática. Pode-se subdividir-se os modelos matemáticos aplicados à epidemiologia em duas grandes classes:

- **Modelos determinísticos:** São modelos onde as populações de indivíduos suscetíveis, infectados e recuperados são descritos como funções de tempo discreto ou de tempo contínuo, por meio de funções diferenciáveis. Desta forma é possível aplicar operações diferenciais sobre as funções que regem as dinâmicas de infecção. Nestes modelos não há a inserção de aleatoriedade, sendo que o processo de infecção da doença evolui de acordo como descrito nas suas regras e funções. São amplamente utilizados para descrever a evolução de uma doença em uma população de grandes proporções.
- **Modelos estocásticos:** São modelos onde considera-se uma população fechada de indivíduos, em que cada um destes pertence a uma classe, podendo ser suscetível, infectado e recuperado. Os indivíduos da população mudam de classe em instantes de tempo aleatórios dentro de um tempo contínuo ou discreto. A aleatoriedade é inerente em processos

infecciosos modelados por estes modelos. São comumente utilizados em populações de pequeno porte.

A integração de modelos teóricos e computacionais é de grande importância no desenvolvimento de modelos epidemiológicos, pois permite o estudo individual de cada parâmetro inerente aos modelos em ambientes simulados, auxiliando na calibração das dinâmicas modeladas e no extenso estudo dos processos dinâmicos em grandes populações de indivíduos e áreas demográficas, sem que seja necessária qualquer intervenção humana no local de estudo. O uso de recursos computacionais na simulação de propagação de doenças através de modelos matemáticos epidemiológicos desperta bastante interesse atualmente, pois a cada dia novas tecnologias são apresentadas e o desempenho dos computadores aumenta gradativamente, permitindo que, cada vez, processos epidemiológicos complexos possam ser simulados computacionalmente em tempos de execução satisfatórios.

## **2.2 Modelos compartimentais: Tipos e Classificações**

Em [Alves e Gagliardi 2006] divide-se a modelagem biológica em dois grandes grupos: o grupo experimental e o grupo de simulação. No grupo experimental são construídos e provados matematicamente modelos experimentais, amplamente utilizados na biologia molecular. O grupo de simulação agrupa os processos e fenômenos que podem ser simulados através de ferramentas computacionais. Em geral a modelagem matemática e computacional é uma alternativa metodológica com o objetivo de entender e manipular mecanismos e processos em objetos de estudo, com um propósito preditivo, no sentido de antecipar comportamentos epidêmicos das doenças modeladas. O cerne principal da modelagem, seja ela matemática ou computacional, é a de abstrair as características mais importantes de um sistema natural, descrevendo-a em termos de equações matemáticas ou métodos computacionais.

A abordagem usualmente empregada na modelagem de doenças infecto-contagiosas é a de diagrama de blocos ou em compartimentos, conhecida como modelagem compartimental. Na modelagem compartimental aplicada à epidemiologia, a população é classificada em categorias, grupos ou compartimentos disjuntos, de acordo com os estados que se queira modelar de uma determinada doença. [Alves e Gagliardi 2006]. Os indivíduos então transitam, de um compartimento à outro, a partir de determinadas probabilidades, que dependem das caracterís-

ticas da dinâmica espaço-temporal da doença. Supondo que  $N(t)$  designa a quantidade total de indivíduos num nível de tempo  $t$  e tomando uma modelagem compartimental, os indivíduos que compõem a população podem ser divididos em classes do tipo suscetíveis,  $S(t)$ ; latentes ou expostos,  $E(t)$ ; infectados,  $I(t)$ , e recuperados ou removidos,  $R(t)$ , de modo que  $S(t) + E(t) + I(t) + R(t) = N(t)$ , observando-se a propriedade de conservatividade na quantidade de indivíduos na população.

Os estados dos indivíduos, isto é, os possíveis estados em que cada indivíduo pode estar em um determinado instante de tempo, comumente modelados utilizando-se modelagem compartimental são:

- **Suscetível  $S$ :** Indica que o indivíduo ainda não contraiu a doença e está apto a adquiri-la.
- **Exposto  $E$ :** Indica que o indivíduo contraiu a doença, mas ainda não pode transmiti-la à outros por um determinado período.
- **Infectado  $I$ :** Indica que o indivíduo está infectado com a doença por um determinado período e é capaz de transmiti-la à outros que estejam no estado suscetível.
- **Recuperado  $R$ :** Indica que o indivíduo já foi infectado com a doença anteriormente e não pode transmiti-la nem contaminar-se novamente em um determinado período.

Considerando-se os estados relacionados com o processo infecto-contagioso, usuais modelos empregados são:

- **Modelo Suscetível-Infectado-Suscetível (SIS):** Modelo empregado em situações em que a doença não apresenta período de latência após a infecção e não confere imunidade, passando os indivíduos do compartimento infectado diretamente para o suscetível.
- **Modelo Suscetível-Infectado-Recuperado (SIR):** Modelo empregado em doenças onde não há período de latência após a infecção e onde os indivíduos infectados podem recuperar-se e adquirir imunidade permanente.
- **Modelo Suscetível-Exposto-Infectado-Recuperado (SEIR):** Utilizado em doenças com período de latência após a infecção e que conferem imunidade permanente.

- **Modelo Suscetível-Exposto-Infectado-Recuperado-Suscetível (SEIRS):** Utilizado em doenças com período de latência após a infecção e que não conferem imunidade permanente, passando os indivíduos do compartimento recuperado ao suscetível novamente.

Neste trabalho é utilizado como base à modelagem de dinâmicas epidemiológicas o modelo SEIRS, visto que é um modelo mais robusto por permitir a modelagem de 4 estados, podendo ser utilizado na modelagem de um amplo conjunto de epidemias. O fluxo de transição de estado dos indivíduos para o modelo SEIRS é como mostrado na Figura 2.1.

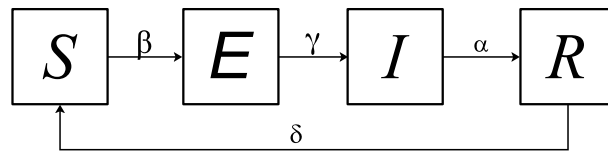


Figura 2.1: Fluxo compartimental no modelo SEIRS

Na Figura 2.1,  $\beta$  designa a taxa de transmissão da doença, que é a taxa de propagação da doença no encontro de indivíduos suscetíveis com infectantes. Após ser infectado o indivíduo entra em um estado de latência, permanecendo em estado exposto por um período designado por  $\gamma$ . O parâmetro  $\alpha$  designa a taxa de recuperação, em que indivíduos infectantes se recuperam da doença, ficando imunes à ela. O taxa de perda de imunidade à doença é caracterizado pelo parâmetro  $\delta$ , que modela a perda de imunidade de indivíduos que estejam recuperados, passando-os novamente para o estado suscetível.

Considerando-se estes modelos compartimentais e suas características, os principais metodologias matemáticas ou computacionais empregadas na sua simulação são:

- **Equações Diferenciais Ordinárias:** Nesta modelagem descreve-se matematicamente o fluxo de transição dos possíveis estados dos indivíduos no modelo SEIRS através de um sistema de equações diferenciais ordinárias não lineares, EDO. Se  $N = S(t) + E(t) + I(t) + R(t)$  designa o tamanho fixo da população de indivíduos, onde suas quantidades nos diferentes estados variam no tempo, de modo que suas quantidades são funções do tempo,  $S = S(t)$ ,  $E = E(t)$ ,  $I = I(t)$  e  $R = R(t)$ , respectivamente, obtém-se os problemas de valor inicial, PVI, para o caso contínuo e discreto, como apresentados em 2.1.

$$\begin{cases} \frac{d(S(t))}{dt} = -\frac{\beta}{N} I(t) S(t) + \delta R(t) \\ S(0) = S_0 \\ \frac{d(E(t))}{dt} = +\frac{\beta}{N} I(t) S(t) - \gamma E(t) \\ E(0) = E_0 \\ \frac{d(I(t))}{dt} = +\gamma E(t) - \alpha I(t) \\ I(0) = I_0 \\ \frac{d(R(t))}{dt} = +\alpha I(t) - \delta R(t) \\ R(0) = R_0 \end{cases} \quad \begin{cases} S^n = S^{n-1} - \frac{\beta}{N} I^{n-1} S^{n-1} + \delta R^{n-1} \\ S^0 = S_0 \\ E^n = E^{n-1} + \frac{\beta}{N} I^{n-1} S^{n-1} - \gamma E^{n-1} \\ E^0 = E_0 \\ I^n = I^{n-1} + \gamma E^{n-1} - \alpha I^{n-1} \\ I^0 = I_0 \\ R^n = R^{n-1} + \alpha I^{n-1} - \delta R^{n-1} \\ R^0 = R_0 \end{cases} \quad (2.1)$$

Nas formulações apresentadas em 2.1,  $S_0$ ,  $E_0$ ,  $I_0$  e  $R_0$  representam, respectivamente, as distribuições iniciais das populações de suscetíveis, expostos, infectantes e recuperados.

- **Autômatos Celulares:** Autômatos celulares, AC, podem ser classificados como sendo modelos de sistemas dinâmicos discretos no espaço e no tempo, que operam sobre um *lattice* contendo uma quantidade finita ou infinita de células. São caracterizados por iterações locais em que cada célula está associada a um estado ou fase por meio de conjunto discreto de valores. A atualização do estado é realizada a partir dos estados das células vizinhas de acordo com regras locais. Autômatos celulares possuem um domínio especificado pelo ambiente ou espaço celular em que estes são operados; um espaço de estados das células em que cada célula do *lattice* possui determinado estado; a vizinhança de cada célula; e a regra de transição local que atua sobre a célula de modo que seu estado pode variar ao ocorrer uma transição. Considerando-se as expressões para o domínio e a regra de transição, e tomando a condição de contorno periódica, o AC é especificado como (2.2):

$$\begin{cases} s_{i,j}^{t+1} = s_{i,j}^t - \beta s_{i,j}^t i_{i,j}^t - s_{i,j}^t \left( \sum_{(\alpha,\beta) \in \mathcal{V}^*} \mu_{\alpha,\beta}^{(i,j)} \frac{N_{i+\alpha,j+\beta}}{N_{i,j}} i_{i+\alpha,j+\beta}^t \right) + \delta r_{i,j}^t \\ e_{i,j}^{t+1} = e_{i,j}^t + \beta s_{i,j}^t i_{i,j}^t + s_{i,j}^t \left( \sum_{(\alpha,\beta) \in \mathcal{V}^*} \mu_{\alpha,\beta}^{(i,j)} \frac{N_{i+\alpha,j+\beta}}{N_{i,j}} i_{i+\alpha,j+\beta}^t \right) - \gamma e_{i,j}^t \\ i_{i,j}^{t+1} = i_{i,j}^t + \gamma e_{i,j}^t - \sigma i_{i,j}^t \\ r_{i,j}^{t+1} = r_{i,j}^t + \sigma i_{i,j}^t - \delta r_{i,j}^t \\ s_{i,j}^0 > 0, \quad e_{i,j}^0 \geq 0, \quad i_{i,j}^0 > 0, \quad r_{i,j}^0 \geq 0, \end{cases} \quad (2.2)$$

A solução de (2.2) é obtida por meio da evolução espaço-temporal dada pela regra de

transição que fornece os estados finais a partir do cálculo dos estados iniciais. A transição se dá num nível de tempo  $t$  para o consecutivo nível de tempo  $t + 1$ .

- **Lattice Gas Cellular Automata:** *Lattice Gas Cellular Automata*, LGCA, são um particular tipo de autômato celular utilizados principalmente na simulação de escoamento e fluxo de fluídos. Neste modelo, especificamente aplicado à epidemiologia, os estados são partículas que se movem no *lattice* à determinadas velocidades. Geralmente são modelados os processos de propagação e colisão. No processo de propagação as partículas são movidas em sua vizinhança à determinadas velocidades, onde somente uma partícula pode ser movimentar em cada conexão. No processo de colisão, regras são utilizadas para determinar o que acontece quando duas ou mais partículas estão em uma mesma posição do *lattice*. Para a construção de um LGCA deve-se especificar um modelo que atenda as características de um autômato celular na fase de propagação e é preciso definir quais regras de colisão que serão utilizadas, dependendo do problema ou dinâmica que será modelada. [McNamara e Zanetti 1988]
- **Sistemas multiagentes:** Sistemas multiagente consistem na especificação de agentes inteligentes, que interagem entre si e com um ambiente, tendo os agentes objetivos bem definidos. Na epidemiologia computacional, agentes baseados em modelos são definidos espaço-temporalmente especificando-se como ocorre a transição do seu estado e seu movimento no ambiente ao longo do tempo. A especificação formal de um agente pode ser realizada por meio de um operador de evolução, que é a composição dos operadores de transição temporal, que realiza a transição do estado do agente considerando sua interação com outros agentes e com o ambiente e de transição espacial, que movimenta os agentes de uma posição à outra dentro do ambiente, considerando seus atributos de conectividade e mobilidade. Sistemas multiagente são interessantes por permitirem a modelagem das classes de indivíduos independentemente umas das outras, por apresentarem comportamentos populacionais, bastante relevante aos estudos epidemiológicos e por sua natureza paralela, permitindo e facilitando o desenvolvimento de sistemas multiagentes paralelos.



## 2.3 Agentes Inteligentes

Segundo [Russel e Norvig 2003] um agente é algo que, por meio de sensores, pode perceber o meio ambiente em que está inserido, podendo agir sobre este meio através de atuadores, sendo capaz de perceber suas próprias ações, mas nem sempre seus efeitos. É capaz de agir de forma autônoma no ambiente com o intuito de atingir seus objetivos. A exemplo de um agente humano, os olhos, nariz e ouvidos seriam sensores e os braços, mãos e boca seriam os atuadores. Agentes robóticos podem contar com sensores eletrônicos, como câmeras e infravermelho e como atuadores diversos motores e braços mecânicos. Agentes de *software* podem receber, por exemplo, pacotes de rede e agir enviando pacotes através da rede. A escolha da ação de um agente em um determinado instante de tempo pode depender de todas as percepções realizadas até aquele momento, e não somente da percepção atual.

O comportamento de um agente pode ser descrito matematicamente através de funções, que mapeiam uma ou várias percepções em uma ação. Um agente e suas funções podem ser implementadas concretamente através de linguagens de programação em programas de agentes. Pode-se ainda definir agentes racionais. Agente racionais são agentes que, para cada possível conjunto de percepções, selecionam uma ação que maximize seu desempenho no ambiente, levando em conta seu conhecimento construído ao longo do tempo. Para a avaliação correta do sucesso de um agente, é indispensável uma medida de desempenho objetiva, que possa representar claramente como avaliar o desempenho do agente, quando atuando no ambiente.

Agentes inteligentes são amplamente utilizados na análise e modelagem matemática e computacional, análise e desenvolvimento de robôs, na área de robótica, em sistemas de extração de informações ou monitoramento de comportamentos ou recursos, em redes de aprendizado e decisão, em jogos eletrônicos, em sistemas comerciais de marketing e propaganda, entre diversas outras aplicações.

Um agente, para ser considerado inteligente, deve ser capaz de desempenhar funções autônomas para atingir seus objetivos. Desta forma, quatro conceitos são importantes afim de determinar se um agente é ou não inteligente: [Wooldridge 2016] e [Wooldridge e Jennings 1995].

- **Autonomia:** Agente inteligentes devem ser capazes de agir sem intervenções diretas de humanos ou outros mecanismos externos, tendo controle sobre suas ações e seu estado.

- **Reatividade:** Agentes inteligentes devem ser capazes de perceber seu ambiente e responder, em um tempo hábil, às mudanças que ocorreram, com o intuito de atingir seus objetivos.
- **Pró-atividade:** Agentes inteligentes devem ser capazes de mostrar um comportamento direcionado aos objetivos, tomando a iniciativa para alcançar seus objetivos.
- **Sociabilidade:** Agentes inteligentes devem ser capazes de interagir com outros agentes para satisfazer seus objetivos.

Um importante componente que interage com o agente é o ambiente. O ambiente pode ser definido como o meio em que os agentes atuam e interagem entre si, sendo que as ações e interações entre os agentes provocam mudanças no ambiente, que são percebidas por estes. Em [Russel e Norvig 2003] e [Wooldridge 2016] são apresentadas diversas classificações para ambientes, destacando-se:

- **Ambiente totalmente observável ou parcialmente observável:** Um ambiente é totalmente observável se os sensores dos agentes que estão inseridos neste ambiente são capazes de prover ao agente o estado completo de todo o ambiente, fornecendo informações relevantes para a escolha de ações. Ambientes totalmente observáveis são interessantes pois os agentes não precisam manter informações internas sobre o estado do ambiente. Um ambiente é parcialmente observável se os sensores dos agentes não são capazes de prover informações do estado do ambiente como um todo, seja por imprecisões ou devido à informações que estão ocultas ou não podem ser obtidas.
- **Determinístico ou estocástico:** Em um ambiente determinístico, seu próximo estado é completamente determinado pelo estado atual e as ações executadas pelos agentes contidos nele. Neste tipo de ambiente, os agentes não precisam se preocupar sobre incertezas, se o ambiente for totalmente observável. Em ambientes estocásticos, o estado futuro do ambiente não depende somente de seu estado atual e das ações de seus agentes, podendo depender de outras informações ou acontecimentos aleatórios. Ambientes parcialmente observáveis geralmente também são ambientes estocásticos.

- **Episódico ou sequencial:** Em um ambiente episódico, as experiências dos agentes são divididas em episódios atômicos, que consistem na percepção do agente e a execução de uma única ação. O episódio futuro não depende de ações ou informações dos episódios passados, sendo que as ações tomadas no episódio atual depende somente dele próprio. Em ambientes sequenciais as decisões atuais podem afetar as decisões que serão tomadas no futuro. Em geral, ambientes episódicos são muito mais simples do que ambientes sequenciais, pois os agentes não precisam planejar ações futuras.
- **Dinâmico ou estático:** Ambientes que podem mudar de estado enquanto os agentes estão tomando suas decisões de ação são chamados de dinâmicos. O ambiente está constantemente perguntando aos agentes que ações realizarão. Se um agente não decidiu sua ação, o ambiente infere que sua ação é de não realizar nada. Em ambientes estáticos os agentes não precisam se preocupar com o estado atual do meio enquanto decidem qual ação tomar e nem com a passagem do tempo. Ambientes estáticos geralmente são mais fáceis de lidar, sob o ponto de vista dos agentes.
- **Discreto ou contínuo:** A distinção entre ambientes contínuos e discretos pode ser realizada através do estado do meio, da forma como o tempo é tratado ou considerado e das percepções e ações dos agentes. Em ambientes discretos, o tempo e seus estados assumem valores finitos e discretos ou são discretizações de informações contínuas, enquanto em ambientes contínuos, essas informações assumem valores infinitos ou contínuos.
- **Agente único ou multiagente:** Diz respeito ao número de agentes interagindo e atuando no ambiente. Em ambientes de agente único, somente um agente está interagindo e agindo sobre o ambiente, enquanto que em ambientes multiagente, há diversos agentes, que tanto interagem entre si, quanto interagem com o ambiente. Ambientes multiagente podem ainda ser classificados em ambientes competitivos, onde os agentes, em grupos ou individualmente, tomam suas decisões levando em conta a maximização de seu desempenho e a minimização do desempenho de outros e em ambientes cooperativos, onde os agentes tomam suas decisões tentando maximizar o desempenho de todos os agentes, como um único grupo.

Quanto aos agentes em particular, pode-se classificá-los em quatro tipos básicos, de

acordo com suas características e aplicações: [Russel e Norvig 2003], [Wooldridge 2016] e [Wooldridge e Jennings 1995].

- **Agentes reativos simples:** Tipo mais simples de agente, com inteligência bastante limitada. São caracterizados pela escolha de ações somente levando em consideração a percepção atual, ignorando seu histórico de percepções. Sua tomada de decisão é baseada em regras do tipo condição-ação, também chamadas de regras de se-então. A condição consiste na percepção atual que o agente recebe de seus sensores e a ação é atuação do agente de acordo com a percepção. São eficazes em ambientes totalmente observáveis onde as decisões corretas podem ser tomadas partindo das percepções atuais do agente. Agentes reativos simples podem randomizar parte de seu processo de decisão afim de reduzir sua probabilidade de entrada em um estado de repetição infinita de decisões, que pode ser extremamente prejudicial em aplicações multiagente.
- **Agentes reativos baseados em modelos:** Agentes especialmente usados em ambientes parcialmente observáveis, que são capazes de manter informações sobre o ambiente, com base em suas percepções, de forma a refletir aspectos atualmente não observáveis do meio. Em geral, esse tipo de agente tem o conhecimento sobre como o ambiente evolui independentemente e como suas ações afetam o meio em que está inserido.
- **Agentes baseados em objetivos:** Nem sempre conhecer o estado atual do ambiente é o suficiente para o agente tomar a decisão correta sobre suas ações. Algumas vezes é necessário que o agente saiba quais estados ou situações são desejáveis. No geral, agentes baseados em objetivos tomam suas decisões levando em conta suas percepções, o conhecimento que adquirem do meio ambiente e seus objetivos. O objetivo pode ser alcançado por simples ações individuais dos agentes ou este pode ter que planejar uma sequência de ações, e pode até mesmo ter que colaborar com outros agentes. Agentes baseados em objetivos são mais flexíveis pois podem ser reconfigurados a qualquer momento de acordo com suas percepções ou mesmo externamente.
- **Agentes baseados em utilidades:** Na maioria dos ambientes, a incorporação de objetivos no agente não é suficiente para modelar comportamentos de alta qualidade. Muitas vezes é necessário agregar diferentes estados não científicos, como sentimentos ou adje-

tivos. Agentes baseados em utilidades fazem uso de uma função de utilidade que mapeia estados ou percepções em um número real, que está associado ao nível de utilidade do agente. Dependendo do nível de utilidade do agente, este pode decidir por diferentes ações estando em um determinado estado, com o objetivo de aumentar seu nível de utilidade de acordo com a realização de ações.

Essas classificações podem derivar a uma classe mais abrangente: os agentes cognitivos. Agentes cognitivos podem ser utilizados em ambientes totalmente desconhecidos e tornar-se mais competentes e eficazes através do ganho de conhecimento. São capazes de aprender e melhorar suas decisões, por meio das experiências e ações tomadas em tempos passados. Podem medir seu desempenho afim de melhorá-lo no futuro. Apresentam um componente importante: o gerador de problemas, que sugere ações ao agente que podem conduzir à obtenção de novas informações e experiências. Podem também apresentar mecanismos de recompensa ou penalidade, que atuam de acordo com seu desempenho e comportamento. Agentes cognitivos são extremamente difíceis de modelar e implementar, por apresentarem comportamentos complexos que mimetizam inteligência, como a de seres humanos ou animais. [Russel e Norvig 2003].

## 2.4 Metodologia de *Bitstring*

Em [Paixão 2012] utilizou-se um modelo *bitstring* ao estudo e simulação da propagação do vírus da dengue. O modelo *bitstring* baseia-se na manipulação direta de *bits* de uma palavra computacional para representar atributos pertencentes aos elementos modelados. A utilização de um modelo *bitstring* justifica-se principalmente pelo fato de que, nesta metodologia, as informações são agrupadas de maneira otimizada, diminuindo o espaço de memória utilizado e, através de operações binárias e de deslocamentos, permitirem a rápida manipulação dos dados. Pode-se ainda, dependendo da dimensão do problema, carregar o programa inteiro para a memória cache do computador, que é uma memória de alta velocidade próxima à CPU, otimizando ainda mais o processo de execução do programa.

Neste trabalho a aplicação da metodologia de *bitstring* justifica-se pelo uso otimizado da memória, que será determinante no ganho de desempenho na posterior paralelização que será realizada utilizando a plataforma CUDA. A redução do uso de memória também é importante por diminuir a quantidade de dados que serão transferidos entre a CPU e a GPU, reduzindo o

tempo necessário à realização de cópias de dados entre os componentes.

Para a implementação de modelos *bitstring* é necessário utilizar uma linguagem de programação que permita operações diretas à *bits*. Em diversas linguagens, a exemplo Fortran e C, é possível manipular seus tipos primitivos, *bit* a *bit*, através de operações da álgebra booleana, como a negação, conjunção, disjunção e deslocamentos. Os tipos primitivos mais interessantes ao uso da técnica de *bitstring* são os inteiros, que geralmente contém 32 *bits* e os reais, que geralmente contém 64 *bits*. O comportamento das operações binárias, ou seja, a saída que fornecem de acordo com suas entradas, que podem ser utilizadas em uma modelagem *bitstring*, é como explicitado a seguir.

- **Operação de negação:** A operação de negação, também conhecida como NOT, é uma operação unária, ou seja, que é aplicada a somente um operando, que consiste em tomar o complemento de um dígito binário. Se o operando de entrada for 0, será retornado 1 e se for 1 será retornado 0. Sua tabela verdade pode ser expressa como em 2.1:

Entrada	Saída
0	1
1	0

Tabela 2.1: Tabela verdade para a operação unária de negação

- **Operação de conjunção:** A operação de conjunção, também conhecida como AND, é uma operação binária, ou seja, que é aplicada a dois operandos, que retorna um *bit* 1 somente se os dois operandos foram 1, e 0 caso contrário. Sua tabela verdade pode ser expressa como em 2.2:

Entrada 1	Entrada 2	Saída
0	0	0
0	1	0
1	0	0
1	1	1

Tabela 2.2: Tabela verdade para a operação binária de conjunção

- **Operação de disjunção inclusiva:** A operação de disjunção inclusiva, também conhecida como OR, é uma operação binária que retorna um *bit* 1 se ao menos um dos operandos for 1, e 0 caso contrário. Sua tabela verdade pode ser expressa como em 2.3:

Entrada 1	Entrada 2	Saída
0	0	0
0	1	1
1	0	1
1	1	1

Tabela 2.3: Tabela verdade para a operação binária de disjunção inclusiva

- **Operação de disjunção exclusiva:** A operação de disjunção exclusiva, também conhecida como XOR, é uma operação binária que retorna um *bit* 1 se somente um dos operandos for 1, e 0 caso contrário. Sua tabela verdade pode ser expressa como em 2.4:

Entrada 1	Entrada 2	Saída
0	0	0
0	1	1
1	0	1
1	1	0

Tabela 2.4: Tabela verdade para a operação binária de disjunção exclusiva

- **Operações de deslocamento:** As operações de *shift* ou de deslocamento consistem em rotacionar um conjunto de *bits* em uma palavra computacional em um determinado fator, à esquerda ou à direita. As operações de deslocamento podem ser realizadas de duas distintas maneiras:
  - **Deslocamentos não circulares:** São deslocamentos onde os *bits* expelidos de uma extremidade são descartados, de modo que novos *bits* com valor 0 são inseridos na outra extremidade da palavra para manter seu tamanho constante. Tomando como exemplo uma palavra de 5 *bits*,  $01001_2$ , e aplicando um deslocamento de fator 1 à direita, a palavra resultante do processo é  $00100_2$ . Note que o *bit* menos significativo foi perdido e um *bit* 0 foi inserido na posição mais significativa da palavra.
  - **Deslocamentos circulares:** São deslocamentos onde os *bits* expelidos de uma extremidade são inseridos na extremidade oposta, conservando os *bits* deslocados dentro da palavra. Tomando novamente como exemplo a palavra  $01001_2$ , se aplicado um deslocamento de fator 1 à direita, a palavra resultante é  $10100_2$ . Note que o *bit* menos significativo, que foi expelido na extremidade à direita da palavra, foi reinserido na extremidade à esquerda da palavra.

Neste trabalho propõe-se o uso da metodologia de *bitstring* para a modelagem dos atributos dos agentes constituintes da população de indivíduos no sistema multiagente. Através das operações citadas anteriormente, serão definidos métodos para a captura e configuração dos atributos dos agentes de forma otimizada, utilizando uma linguagem de programação que seja apropriada às operações aqui descritas.



# Capítulo 3

## Metodologias Computacionais

### 3.1 Introdução

Neste capítulo serão apresentadas as metodologias computacionais utilizadas à realização deste trabalho. Serão discutidas as modelagens realizadas à representação dos indivíduos no sistema multiagente proposto, expondo-as em termos de operadores e ao uso da metodologia em *bitstring*, as linguagens de programação e as motivações de seus usos, as estruturas de dados desenvolvidas para a apropriada aplicação da modelagem *bitstring* e paralelização em CUDA e estratégias de implementação do sistema multiagente. Em sequência serão discutidas questões relativas às API para programação paralela OpenMP e CUDA, discutindo suas principais características e aplicações. Por fim será apresentado o *software* especialmente desenvolvido à manipulação de informações georreferenciadas, que será utilizado como apoio às operações de configuração de parâmetros e do *lattice* e à visualização de resultados das simulações que serão executadas.

### 3.2 Modelagem em Operadores aos Agentes

Os modelos apresentados em 2.2 serviram à fundamentação do modelo baseado em agentes, que será desenvolvido e utilizado na implementação e execução de simulações computacionais. A modelagem empregada para simular o espalhamento de hipotética doença de transmissão direta em indivíduos considera agentes baseados em modelos que são definidos espaço-temporalmente especificando-se como ocorre a transição do seu estado num intervalo de tempo e seu movimento no ambiente, de uma posição para outra no passo de tempo. Um passo de tempo é especificado como um ciclo de transição.

Um agente  $\chi(t)$  é definido espaço-temporalmente especificando-se como ocorre a transição do seu estado num intervalo de tempo  $t$  e seu movimento no espaço. O estado do agente  $\chi(t)$  é especificado através do seguinte conjunto de atributos, como em 3.1.

$$\chi(t) \equiv (L, X, Y, C, E) \quad (3.1)$$

Cujos significados dos identificadores dos atributos do estado interno do agente  $\chi(t)$  são como:

- Lote ( $L$ ): Identificador do lote que o agente  $\chi(t)$  se encontra.
- Posição em x ( $X$ ): Coordenada  $x$  da posição do agente  $\chi(t)$  no lote.
- Posição em y ( $Y$ ): Coordenada  $y$  da posição do agente  $\chi(t)$  no lote.
- Contador de controle ( $C$ ): Contador de ciclos que controla os períodos de transição entre os estados do agente.
- Estado ( $E$ ): Identificador do estado atual do agente  $\chi(t)$ .

De maneira que um agente é designado por  $\chi(t) \equiv (L, X, Y, C, E)$ .

A especificação formal de um agente é realizada através de um operador de evolução que define o estado atual do agente, quando interagindo com o ambiente. Esse operador decorre da composição entre os operadores de transição temporal, que realiza uma transição do estado interno do agente considerando-se sua interação com outros agentes e com o ambiente, e o operador de transição espacial, que movimenta o agente de sua posição para outra, considerando-se os atributos de conectividade e mobilidade.

À dinâmica populacional são considerados três tipos de operações:

- **Movimentação:** Nas operações de movimentação, os indivíduos são movimentados dentro de um ambiente virtual com topologia matricial através de suas vizinhanças de Moore às posições escolhidas aleatoriamente, respeitando os limites do ambiente.
- **Contato:** Nas operações de contato ocorre, probabilisticamente, a transmissão da doença através dos indivíduos infectados para os indivíduos suscetíveis que ocupam uma mesma posição no ambiente.

- **Transição de estados:** Nas transições de estados, ocorre a passagem de estados dos indivíduos de expostos para infectantes, de infectantes para recuperados e de recuperados para suscetíveis.

Tais operações são realizadas na sequência em que foram apresentadas e uma vez a cada ciclo, que consiste na aplicação dos operadores sobre a população de indivíduos e geração de arquivos de saída. Uma simulação é composta por uma determinada quantidade de ciclos.

Cada agente implementa uma operação de evolução  $\lambda$  que atualiza o estado atual do agente quando interagindo com o ambiente, definido como  $\lambda(\chi(t)) = \sigma(\mu(\chi(t)))$ , que decorre da composição entre os operadores  $\mu$  e  $\sigma$ . O operador de transição espacial,  $\mu$ , movimenta o agente de sua posição considerando-se os atributos de conectividade e mobilidade, e o operador de transição temporal  $\sigma$  realiza a transição do estado interno do agente considerando-se sua interação com outros agentes e o ambiente.

O operador espaço-temporal  $\lambda(\chi(t))$  realiza as operações do agente  $\chi(t)$  movimentando-o da posição  $(i, j)$  para uma posição  $(\xi, \eta)$  no ciclo de tempo atual,  $t$ , para o ciclo de tempo,  $t + 1$ . Formalmente o estado resultante das operações no agente  $\chi(t)$  é representado como em 3.2:

$$\lambda(\chi(t)^{(i,j)}) \equiv \sigma(\mu(\chi(t+1))^{(\xi,\eta)}) \quad (3.2)$$

### 3.3 Modelagem em *Bitstring* aos Agentes

A modelagem *bitstring* realizada à representação do agente é baseada na manipulação diretas dos *bits* em uma palavra computacional, que é capaz de caracterizar sem ambigüidade a especificação do agente  $\chi(t) = (L, X, Y, C, E)$ , com identificador do lote,  $L$ , coordenada  $x$  da posição,  $X$ , coordenada  $y$  da posição,  $Y$ , contador de controle,  $C$ , e estado,  $E$ .

Ao emprego de um modelo em *bitstring* é necessário utilizar uma linguagem de programação que dê suporte apropriado às operações diretas com *bits*. À implementação do sistema multiagente, objetivando alcançar as metas definidas neste trabalho, é proposta a utilização da linguagem de programação  $C$ , que provê suporte aos propósitos de modelagem. O tipo de dado inteiro em  $C$ , *int*, contém 32 *bits*, sendo suficiente à especificação da formulação para o agente  $\chi(t)$  em termos de *bits*, sendo 32 a quantidade total de *bits* na palavra. O uso do tipo de dado *unsigned int* descarta a preservação de sinal na realização de *shifts* ou deslocamentos, evitando

a introdução de erros na manipulação dos *bits* mais significativos em operações de captura e configuração de atributos de um agente.

À identificação do lote atual,  $L$ , do agente, considera-se que o identificador não ultrapasse 64 valores. À modelagem da posição  $X, Y$  de um agente, considera-se que as quantidades de linhas,  $\#L$ , e de colunas,  $\#C$ , do ambiente, são limitadas por  $\max(\#L \times \#C) = (512 \times 512)$ , onde  $\max$  indica o máximo possível de linhas e de colunas que podem ser alocadas à essa escolha. À modelagem do contador de controle,  $C$ , de um agente, considera-se que a quantidade de ciclos em que um agente fica no estado exposto, infectado ou recuperado, não ultrapassem 64 ciclos de simulação. À modelagem do estado  $E$  de um agente, considerando a adoção do modelo compartimental tipo SEIRS, existem somente quatro distintos estados.

Observe que  $512_{10} = 2^9$ ,  $64_{10} = 2^6$  e  $4_{10} = 2^2$ , sendo suficiente uma palavra que comporte um quantidade de 32 *bits* para armazenar a especificação do agente  $\chi(t) = (L, X, Y, C, E)$  em *bits*. A utilização de faixas maiores à representação dos atributos de um agente demandariam outros tipos de dados em  $C$ , com maior quantidade de *bits*. A representação da tira proposta à modelagem dos agentes é como ilustrado na Figura 3.1.



Figura 3.1: Representação da tira do agente  $\chi$

Na Figura 3.1, os elementos do identificador do lote  $L$  estão nas posições 31 a 26. Os elementos das coordenadas  $X$  e  $Y$  da posição do agente no lote estão nas posições 25 a 17 e 16 a 8, respectivamente. Os elementos do contador de controle  $C$  e do estado  $E$  estão nas posições 7 a 2 e 1 a 0, respectivamente. Considerando as escolhas realizadas à sua modelagem, um agente qualquer do modelo pode ser representado em *bits* como em 3.3.

$$\chi(L, X, Y, C, E) \equiv (l_{31}, \dots, l_{26}; x_{25}, \dots, x_{17}; y_{16}, \dots, y_8; c_7, \dots, c_2; e_1, \dots, e_0) \quad (3.3)$$

sendo que os elementos  $l_i$ ,  $x_i$ ,  $y_i$ ,  $c_i$  e  $e_i$  possuem o valor 0 ou 1 e seus índices subscritos indicam que cada *bit* pode variar nos respectivos intervalos que estão especificados.

Para identificar o lote e as coordenadas da posição do agente  $\chi(L, X, Y, C, E) = \chi(t)$  no ambiente, assim como seu estado interno e o registro de controle de ciclos às transições de

estados, essas características são definidas como em 3.4, para os ciclos  $t$  e  $t+1$ , respectivamente como:

$$\begin{cases} L(t+1) = \left( \left( \chi(L, X, Y, C, E) \right) \ll_{arth} (Pl) \right) \gg_{arth} (C\#l) \\ X(t+1) = \left( \left( \chi(L, X, Y, C, E) \right) \ll_{arth} (Px) \right) \gg_{arth} (C\#x) \\ Y(t+1) = \left( \left( \chi(L, X, Y, C, E) \right) \ll_{arth} (Py) \right) \gg_{arth} (C\#y) \\ C(t+1) = \left( \left( \chi(L, X, Y, C, E) \right) \ll_{arth} (Pc) \right) \gg_{arth} (C\#c) \\ E(t+1) = \left( \left( \chi(L, X, Y, C, E) \right) \ll_{arth} (Pe) \right) \gg_{arth} (C\#e) \end{cases} \quad (3.4)$$

em que  $Pl$ ,  $Px$ ,  $Py$ ,  $Pc$  e  $Pe$  designam, respectivamente, as quantidades de *bits* posteriores à cada campo. A quantidade de *bits* posteriores à cada campo é a quantidade de *bits* existentes que são mais significativos àqueles do campo em questão. Os termos  $C\#l$ ,  $C\#x$ ,  $C\#y$ ,  $C\#c$  e  $C\#e$  designam, respectivamente, as quantidades de *bits* complementares à cada campo. A quantidade de *bits* complementares à cada campo é a diferença entre a quantidade de *bits* total da palavra e a quantidade de *bits* do campo. As operações  $\ll_{arth}$  e  $\gg_{arth}$  designam, respectivamente, deslocamentos não circulares à esquerda e à direita na palavra, na quantidade de *bits* indicado.

Na modelagem realizada, as especificações às quantidades são  $Pl = 0$ ,  $Px = 6$ ,  $Py = 15$ ,  $Pc = 24$ ,  $Pe = 30$ ,  $C\#l = 26$ ,  $C\#x = 23$ ,  $C\#y = 23$ ,  $C\#c = 26$  e  $C\#e = 30$ , de modo que 3.4 é então reescrita para os ciclos  $t$  e  $t+1$  designados como *old* e *new*, respectivamente, como em 3.5.

$$\begin{cases} L_{new} := \left( \chi(L_{old}, X, Y, C, E) \ll_{arth} (0) \right) \gg_{arth} (26) \\ X_{new} := \left( \chi(L, X_{old}, Y, C, E) \ll_{arth} (6) \right) \gg_{arth} (23) \\ Y_{new} := \left( \chi(L, X, Y_{old}, C, E) \ll_{arth} (15) \right) \gg_{arth} (23) \\ C_{new} := \left( \chi(L, X, Y, C_{old}, E) \ll_{arth} (24) \right) \gg_{arth} (26) \\ E_{new} := \left( \chi(L, X, Y, C, E_{old}) \ll_{arth} (30) \right) \gg_{arth} (30) \end{cases} \quad (3.5)$$

As operações definidas em 3.4 e 3.5 capturam informações armazenadas nos agentes, como especificado em 3.1.

Também são necessárias outras operações para manipular os atributos dos agentes, além daquelas definidas em 3.5. Essas operações têm por finalidade setar no agente as informações necessárias às atualizações do identificador de seu lote, posição, estado e contador. São definidas como em 3.6, sendo que  $\ll_{circ}$  e  $\gg_{circ}$  designam, respectivamente, deslocamentos circulares

à esquerda e à direita na palavra, na quantidade de *bits* indicado.

$$\begin{cases} \ll_{circ} := \left( z(t+1) \ll_{arth} \text{shift} \right) \mid \left( z(t+1) \gg_{arth} (N\_BITS - \text{shift}) \right) \\ \gg_{circ} := \left( z(t+1) \gg_{arth} \text{shift} \right) \mid \left( z(t+1) \gg_{arth} (N\_BITS - \text{shift}) \right) \end{cases} \quad (3.6)$$

sendo que  $z(t+1)$  denota ou  $x(t+1)$  ou  $y(t+1)$ ,  $N\_BITS$  denota a quantidade de bits totais,  $\text{shift}$  denota a quantidade de *bits* para os deslocamentos e  $\mid$  denota a operação lógica OR, "ou inclusivo", que é realizada *bit a bit* à palavra. Assim, a especificação às quantidades adotadas na modelagem, 3.6 é reescrita como em 3.7.

$$\begin{cases} \ll_{circ} := \left( z(t+1) \ll_{arth} \text{shift} \right) \mid \left( z(t+1) \gg_{arth} (32 - \text{shift}) \right) \\ \gg_{circ} := \left( z(t+1) \gg_{arth} \text{shift} \right) \mid \left( z(t+1) \gg_{arth} (32 - \text{shift}) \right) \end{cases} \quad (3.7)$$

Subsequentemente são utilizadas as operações definidas em 3.4 e em 3.6, ou concretamente, as definidas em 3.5 e 3.7, para implementar o operador espaço-temporal  $\lambda(\chi(t))$  que movimenta o agente da posição  $(i, j)$  para uma posição  $(\xi, \eta)$  no ciclo de tempo atual,  $t$  (*old*), para o ciclo de tempo,  $t+1$  (*new*). A combinação dessas operações resulta que a identificação do lote, as coordenadas  $x$  e  $y$  da posição, o controle e estado do agente são atualizadas, respectivamente, para  $\lambda(\chi(t))$  por 3.8.

$$\begin{cases} \chi(L(t+1), X(t), T(t), C(t), E(t)) := \left[ \left( \left( \chi(X(t), Y(t), C(t), E(t)) \gg_{circ} (Al) \right) \gg_{arth} (\#l) \right) \ll_{arth} (\#l) \right] \mid L(t+1) \ll_{circ} (Al) \\ \chi(L, X(t+1), Y(t), C(t), E(t)) := \left[ \left( \left( \chi(X(t), Y(t), C(t), E(t)) \gg_{circ} (Ax) \right) \gg_{arth} (\#x) \right) \ll_{arth} (\#x) \right] \mid X(t+1) \ll_{circ} (Ax) \\ \chi(L, X(t), Y(t+1), C(t), E(t)) := \left[ \left( \left( \chi(X(t), Y(t), C(t), E(t)) \gg_{circ} (Ay) \right) \gg_{arth} (\#y) \right) \ll_{arth} (\#y) \right] \mid Y(t+1) \ll_{circ} (Ay) \\ \chi(L, X(t), Y(t), C(t+1), E(t)) := \left[ \left( \left( \chi(X(t), Y(t), C(t), E(t)) \gg_{circ} (Ac) \right) \gg_{arth} (\#c) \right) \ll_{arth} (\#c) \right] \mid C(t+1) \ll_{circ} (Ac) \\ \chi(L, X(t), Y(t), C(t), E(t+1)) := \left[ \left( \left( \chi(X(t), Y(t), C(t), E(t)) \gg_{circ} (Ae) \right) \gg_{arth} (\#e) \right) \ll_{arth} (\#e) \right] \mid E(t+1) \ll_{circ} (Ae) \end{cases} \quad (3.8)$$

em que  $Al$ ,  $Ax$ ,  $Ay$ ,  $Ac$  e  $Ae$  designam, respectivamente, as quantidades de *bits* anteriores à cada campo. A quantidade de *bits* anteriores à cada campo é a quantidade de *bits* existentes que são menos significativos àqueles do campo em questão. Os termos  $\#l$ ,  $\#x$ ,  $\#y$ ,  $\#c$  e  $\#e$  designam, respectivamente, as quantidades de *bits* de cada campo.

Como  $Al = 26$ ,  $Ax = 17$ ,  $Ay = 8$ ,  $Ac = 2$ ,  $Ae = 0$ ,  $\#l = 6$ ,  $\#x = 9$ ,  $\#y = 9$ ,  $\#c = 6$  e

$\#e = 2$ , a especificação às quantidades adotadas na modelagem, 3.8 é reescrita como em 3.9.

$$\left\{ \begin{array}{l} \chi(L_{new}, X_{old}, Y_{old}, C_{old}, E_{old}) := \left[ \left( \left( \chi(L_{old}, X_{old}, Y_{old}, C_{old}, E_{old}) \gg_{circ} (26) \right) \gg_{arth} (6) \right) \ll_{arth} (6) \right] | L_{new} \ll_{circ} (26) \\ \chi(L_{old}, X_{new}, Y_{old}, C_{old}, E_{old}) := \left[ \left( \left( \chi(L_{old}, X_{old}, Y_{old}, C_{old}, E_{old}) \gg_{circ} (17) \right) \gg_{arth} (9) \right) \ll_{arth} (9) \right] | X_{new} \ll_{circ} (17) \\ \chi(L_{old}, X_{old}, Y_{new}, C_{old}, E_{old}) := \left[ \left( \left( \chi(L_{old}, X_{old}, Y_{old}, C_{old}, E_{old}) \gg_{circ} (8) \right) \gg_{arth} (9) \right) \ll_{arth} (9) \right] | Y_{new} \ll_{circ} (8) \\ \chi(L_{old}, X_{old}, Y_{old}, C_{new}, E_{old}) := \left[ \left( \left( \chi(L_{old}, X_{old}, Y_{old}, C_{old}, E_{old}) \gg_{circ} (2) \right) \gg_{arth} (6) \right) \ll_{arth} (6) \right] | C_{new} \ll_{circ} (2) \\ \chi(L_{old}, X_{old}, Y_{old}, C_{old}, E_{new}) := \left[ \left( \left( \chi(L_{old}, X_{old}, Y_{old}, C_{old}, E_{old}) \gg_{circ} (0) \right) \gg_{arth} (2) \right) \ll_{arth} (2) \right] | E_{new} \ll_{circ} (0) \end{array} \right] \quad (3.9)$$

Nas Tabelas 3.1, 3.2, 3.3 e 3.4 são apresentados sistematicamente os valores relativos às quantidades de *bits* dos tamanhos, complementos, posteriores e anteriores de cada atributo do agente  $\chi$ .

Tabela 3.1: Tabela de tamanhos

Atributo	Valor
Identificador do Lote $L$	6
Coordenada $X$	9
Coordenada $Y$	9
Contador de Controle $C$	6
Estado $E$	2

Tabela 3.2: Tabela de complementares

Atributo	Valor
Identificador do Lote $L$	26
Coordenada $X$	23
Coordenada $Y$	23
Contador de Controle $C$	26
Estado $E$	30

Tabela 3.3: Tabela de posteriores

Atributo	Valor
Identificador do Lote $L$	0
Coordenada $X$	6
Coordenada $Y$	15
Contador de Controle $C$	24
Estado $E$	30

Tabela 3.4: Tabela de anteriores

Atributo	Valor
Identificador do Lote $L$	26
Coordenada $X$	17
Coordenada $Y$	8
Contador de Controle $C$	2
Estado $E$	0

Para completar a modelagem é suficiente detalhar e relacionar os identificadores dos lotes e dimensões dos lotes, a faixa de variação do controle, e os possíveis tipos de estados do agente na base 10, com tais características na base 2, para determinar sem ambigüidade os elementos de 3.1.

Os identificadores dos lotes têm variação de  $(0)_{10} = (000\ 000)_2$  até  $(63)_{10} = (111\ 111)_2$ , totalizando 64 possíveis valores. Os lotes têm dimensão máxima de  $512_{10} = 2^9$  linhas por  $512_{10} = 2^9$  colunas, sendo possível variarem numa representação matricial da posição  $(0;0)_{10} = (000\ 000\ 000; 000\ 000\ 000)_2$  até a posição  $(511; 511)_{10} =$

$(111\ 111\ 111; 111\ 111\ 111)_2$ , totalizando os 512 possíveis valores. Semelhantemente, faz-se uma representação matricial ao controle para que sua variação ocorra de  $(0)_{10} = (000\ 000)_2$  até  $(63)_{10} = (111\ 111)_2$ , totalizando os 64 possíveis valores. Os estados do agente são setados como  $(0, 0)_2$  para o suscetível,  $S$ ,  $(0, 1)_2$  para o exposto,  $E$ ,  $(1, 0)_2$  para o infectante,  $I$ , e  $(1, 1)_2$  para o recuperado,  $R$ .

### **3.4 Estruturas de Dados, Linguagens e Estratégias de Implementação**

### **3.5 CUDA e OpenMP**

### **3.6 SIMULA**



# Capítulo 4

## Soluções

### 4.1 Introdução

### 4.2 Implementação Padrão com CUDA e OpenMP

### 4.3 Implementação *Bitstring* com CUDA e OpenMP

### 4.4 Discussões Qualitativas, Quantitativas, Eficiência e Acurácia

# **Capítulo 5**

## **Resultados e Discussões**

### **5.1 Introdução**

### **5.2 Cases: Discutir as Simulações no Espaço Geográfico Escolhido.**

# Referências Bibliográficas

- [Alves e Gagliardi 2006]ALVES, D.; GAGLIARDI, H. F. *Técnicas de Modelagem de Processos Epidêmicos e Evolucionários*. 1. ed. São Paulo: Sociedade Brasileira de Matemática Aplicada e Computacional, 2006.
- [Daley e Gani 1999]DALEY, D. J.; GANI, J. *Epidemic Modelling - An Introduction*. 1. ed. Reading: Cambridge University, 1999.
- [McNamara e Zanetti 1988]MCNAMARA, G. R.; ZANETTI, G. Use of the boltzmann equation to simulate lattice-gas automata. *Physical Review Letters*, EUA, v. 61, n. 20, p. 2332–2335, 1988.
- [Medronho et al. 2008]MEDRONHO, R. A. et al. *Epidemiologia*. 2. ed. Reading: Atheneu, 2008.
- [Paixão 2012]PAIXÃO, C. A. *Modelo de Bitstring para Estudo da Propagação da Dengue*. Tese (Tese de Doutorado) — Universidade Federal de Lavras, Lavras, MG, 2012.
- [Russel e Norvig 2003]RUSSEL, S.; NORVIG, P. *Artificial Intelligence - A Modern Approach*. 2. ed. Reading: Prentice Hall, 2003.
- [Wooldridge 2016]WOOLDRIDGE, M. *Intelligent Agents*. 2016. Consultado na INTERNET: <http://www.cs.ox.ac.uk/people/michael.wooldridge/pubs/mas99.pdf>, 2016.
- [Wooldridge e Jennings 1995]WOOLDRIDGE, M.; JENNINGS, N. R. Intelligent agents - theory and practice. *The Knowledge Engineering Review*, England, v. 10, n. 2, p. 115–151, 1995.
- [Yang 2001]YANG, H. M. *Epidemiologia Matemática - Estudo dos Efeitos da Vacinação em Doenças de Transmissão Direta*. 1. ed. Reading: Unicamp, 2001.