Gleyce Alves Pereira da Silva

Processo de difusão discreto sobre grafo de estados como modelo da dinâmica de espalhamento de doenças em uma população: o caso COVID-19

Recife

Fevereiro de 2023

Universidade Federal Rural de Pernambuco
Pró-Reitoria de Pesquisa e Pós-Graduação
Programa de Pós-Graduação em Biometria e Estatística Aplicada

Processo de difusão discreto sobre grafo de estados como modelo da dinâmica de espalhamento de doenças em uma população: o caso COVID-19

Dissertação julgada como adequada para obtenção do título de Mestre em Biometria e Estatística Aplicada.

Área de concentração: Biometria e Estatística Aplicada

Orientador: Dr. Cláudio Tadeu Cristino

Recife

Fevereiro de 2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal Rural de Pernambuco
Sistema Integrado de Bibliotecas
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

S586p Silva, Gleyce Alves Pereira da Silva
Processo de difusão discreto sobre grafo de estados como modelo da dinâmica de espalhamento de doenças em uma população: o caso COVID-19 / Gleyce Alves Pereira da Silva Silva. - 2023.
65 f.

Orientador: Claudio Tadeu Cristino.
Inclui referências e anexo(s).

Dissertação (Mestrado) - Universidade Federal Rural de Pernambuco, Programa de Pós-Graduação em Biometria e Estatística Aplicada, Recife, 2023.

1. Grafos. 2. Espaço de Estados. 3. Processos de Difusão. 4. Modelagem. 5. Modelos SIR.. I. Cristino, Claudio Tadeu, orient. II. Título

CDD 519.5

Universidade Federal Rural de Pernambuco
Pró-Reitoria de Pesquisa e Pós-Graduação
Programa de Pós-Graduação em Biometria e Estatística Aplicada

Processo de difusão discreto sobre grafo de estados como modelo da dinâmica de espalhamento de doenças em uma população: o caso COVID-19

Gleyce Alves Pereira da Silva

Dissertação julgada como adequada para obtenção do título de Mestre em Biometria e Estatística Aplicada.

Orientador:

Dr. Cláudio Tadeu Cristino
Orientador

Banca examinadora:

Dr. Antonio Samuel Alves da Silva
Universidade Federal Rural de Pernambuco.

Dr.a Silvana Bocanegra
Universidade Federal de Pernambuco.

Este trabalho é dedicado aos meus pais, Ana e Timoteo,
e minhas irmãs, que foram meu alicerce
durante essa jornada.

Agradecimentos

Agradeço aos meus pais, Ana e Timoteo, por me ensinarem a importância dos estudos e me proporcionarem uma criação com amor e incentivo para que eu continuasse nesse caminho. Às minhas irmãs, Geovanna e Gleyciane, agradeço por serem meu alicerce durante toda essa jornada. Aos amigos, meus mais profundos agradecimentos.

Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES por me disponibilzarem uma bolsa de estudos a qual tornou possível a relização do mestrado. À Universidade Federal Rural de Pernambuco - UFRPE, onde passei 7 anos, incluindo graduação e mestrado, minhas mais sinceras gratidões por essa instituição.

Ao meu orientador, o professor Dr. Cláudio Tadeu Cristino, agradeço por toda sua prestabilidade e dedicação durante os anos, cuja orientação foi parte primordial para a finalização desse trabalho e cujo o incentivo durante a graduação me levou ao ingresso no mestrado e continuidade no meio acadêmico.

"A ciência é mais que um corpo de conhecimento,
é uma forma de pensar,
uma forma cética de interrogar o universo
com pleno conhecimento da falibilidade humana."
(Carl Sagan)

Resumo

Dada a incerteza em que operam os sistemas reais, principalmente quando estes envolvem, por suas naturezas, ações humanas imprevisíveis ou avarias de máquinas, a modelagem matemática e computacional dos fenômenos reais em várias situações deve ser feita de maneira cautelosa. Por vezes, são usados modelos determinísticos, que contribuem para a compreensão do comportamento da dinâmica do sistema estudado, como uma primeira aproximação. Por outro lado, muitos sistemas podem ser descritos por modelos probabilísticos, aproveitando certas características de regularidade que eles apresentem. Assim, pode-se recorrer à Teoria dos Processos Estocásticos como uma forma de tratar estes fenômenos, mais particularmente, à utilização de processos markovianos como modelo primordial. Dado um sistema dinâmico, cuja dinâmica pode ser dada no tempo contínuo ou discreto, faz-se necessário um estudo de sua evolução de estados ao longo do tempo. Em algumas aplicações, a distinção entre sistemas contínuos e discretos não é crítica, e a escolha se dá por conveniência. Esse trabalho busca a descrição de um algoritmo que seja capaz de representar uma discretização do processo de espalhamento de uma doença infecciosa, especificamente a COVID-19, pela população brasileira considerada como sistema com três estados possíveis: Suscetível, Infectado ou Recuperado. Esta definição do grafo é feita de maneira natural, formando-se um grafo conexo, denominado grafo de estados mistos, usado para se estudar a dinâmica de estados ao longo do tempo. Cada vértice deste grafo representa um estado da população, ou seja, o percentual de cada um dos grupos descritos. Podendo assim, dado o *input*, posição atual da doença (número de infectados e número de recuperados), gerar o *output* adequado (projeção para estados futuros), respondendo assim as questões pertinentes ao sistema.

Palavras-chaves: Grafos. Espaço de Estados. Processos de Difusão. Modelagem. Modelos SIR.

Abstract

Given the uncertainty in which real systems operate, especially when it involves, by its nature, unpredictable human actions or machine malfunctions. It becomes necessary to search for deterministic models, which contribute to the understanding of the dynamic behavior of a system, at the basic level. Such systems can be described by probabilistic models, taking advantage of certain features of regularity that they exhibit. Thus, one can resort to Stochastic Processes as a way to treat these phenomena quantitatively, depending on certain characteristics one can resort to Markov Processes. Given a dynamical system, whose dynamics can be given in continuous or discrete time, a study of its evolution of states over time is necessary. In some applications, the distinction between continuous and discrete systems is not critical, and the choice is made for convenience. This work seeks the description of an algorithm that is able to discretize the system studied, in a natural way, thus forming a connected graph, in order to study the dynamics of states of this same graph over time. Thus, given the input, it can generate the appropriate output, thus answering the questions pertinent to the system.

Keywords: Graphs. State Spaces. Diffusion process. Modelling. SIR Model.

Lista de Figuras

Sumário

1 Introdução

1.1 Sistemas e Modelagem Matemático Computacional

Um sistema (complexo) pode ser definido como um conjunto para o qual podem-se descrever comportamentos individuais e coletivos, e que podem ser determinísticos ou aleatórios. Nesse último caso, O comportamento do sistema não pode ser explicado apenas observando-se suas propriedades gerais, mas e baseando-se na realidade (dados observados) que surge a partir das interações dos seus constituintes (WOLFRAM, 2018).

Os sistemas reais podem ser descritos, geralmente, por modelos probabilísticos, aproveitando certas características de regularidade que eles apresentam, podendo-se assim recorrer a Processos Estocásticos como uma forma de tratar quantitativamente estes fenômenos, a depender de certas características pode-se recorrer a Processos Markovianos. Os sistemas podem ser tratados através de equações diferenciais parciais, cálculo variacional, equações diferenciais ordinárias, álgebra, entre outros, o modelo de tratamento vai variar de acordo com a característica do sistema em questão. Zuben e Castro (2003) afirmam, ainda, que encontrar a descrição matemática de um sistema equivale a determinar um conjunto de relações matemáticas entre os atributos dos objetos que compõem o sistema, a partir do conhecimento das relações entre estes objetos e entre os atributos de um mesmo objeto. Podendo-se assim associar a um sistema dinâmico uma entidade matemática precisa, a dinâmica pode ser dada no tempo contínuo ou discreto, e sua evolução no tempo é não antecipativa, ou seja a cada instante de tempo $t \in T$ (onde $T \in \mathbb{R}^+_{\{0\}}$ ou $\mathbb{Z}^+_{\{0\}}$ se o sistema operar no tempo contínuo ou discreto respectivamente) o sistema dinâmico recebe alguma entrada $u(t) \in U$ e emite alguma saída $y(t) \in Y$, (onde $U, Y \in \mathbb{R}^n$, em que n não é necessariamente o mesmo para U e Y).

Todo sistema opera com um conjunto de valores das grandezas físicas, denominado como *estado*, que é necessário e suficiente para caracterizar univocamente a situação física deste sistema. Zuben e Castro (2003) afirmam que em uma abordagem de entrada-saída, pode-se dizer que o estado é uma parametrização dos pares de entrada-saída. A evolução do estado de um sistema dinâmico ao longo do tempo é denominada trajetória (ou fluxo, ou órbita) no espaço de estados, sendo determinada pela função de transição de estado. Um sistema dinâmico é dito de tempo contínuo se e somente se a variável temporal t

assume valores contínuos, e é dito de tempo discreto se e somente se a variável temporal t assume valores discretos. Da mesma maneira um sistema dinâmico é dito de estado contínuo se e somente se o espaço de estados X é contínuo, e dito de estado discreto se e somente se o espaço de estados X é discreto.

Por vezes se faz necessário discretizar o sistema para tornar os cálculos dos valores mais fáceis de serem feitos, por vezes essa discretização ocorre de maneira natural, sistemas de tempo contínuo correspondem, por exemplo, aos modelos da física clássica, enquanto que sistemas de tempo discreto surgem, por exemplo, sempre que computadores digitais fazem parte do sistema. Em algumas aplicações, a distinção entre sistemas contínuos e discretos não é crítica, e a escolha se dá por conveniência. Esse trabalho busca a descrição de um algoritmo que seja capaz de discretizar o sistema estudado, de maneira natural, formando assim um grafo conexo, para poder estudar a dinâmica de estados nesse grafo ao longo do tempo. Podendo assim, dado o *input*, gerar o *output* adequado, respondendo assim as questões pertinentes ao sistema.

1.2 COVID-19

Em Dezembro de 2019, na cidade de Wuhan na China, foi constatado um espalhamento em escala global da doença, denominada de COVID-19, causada pelo novo Coronavírus (SARS-CoV-2). A pandemia do COVID-19 teve um impacto significativo na sociedade brasileira, com milhões de casos e milhares de mortes registradas. O país enfrentou diversos desafios no combate à doença, incluindo a falta de equipamentos de proteção individual, leitos hospitalares e medicamentos.

Desde então, diversos trabalhos foram produzidos na área, gerando um grande volume de publicações. As abordagens mais utilizadas para o estudo da propagação e dinâmica do COVID-19 são a modelagem matemática, sendo um dos modelos mais utilizados o modelo SIR (Susceptíveis-Infectados-Recuperados), capaz de prever a disseminação da doença com base em informações sobre a população, a taxa de transmissão e outros fatores relevantes.

No entanto, a abordagem de modelagem matemática com base em equações diferenciais pode ser desafiadora para compreender e aplicar em algumas situações. A abordagem de modelagem matemática proposta, baseada em transformar o espaço de estado do sistema em um grafo conexo, pode oferecer uma alternativa interessante e complementar ao modelo SIR. Ao representar o sistema como um grafo, é possível estudar a dinâmica de estados e interações entre indivíduos ou grupos de indivíduos ao longo do tempo.

Essa abordagem pode ser particularmente útil para entender a propagação do COVID-19 em diferentes contextos, como em ambientes urbanos ou rurais, em diferentes populações e em diferentes momentos da pandemia. A modelagem baseada em grafos pode ser aplicada em combinação com outras técnicas, como a análise de redes sociais, para entender melhor a dinâmica da propagação da doença e identificar possíveis pontos de intervenção para a prevenção e controle.

No entanto, a modelagem matemática por si só não é suficiente para prevenir e controlar a doença. É fundamental que as estratégias de prevenção e controle sejam baseadas em evidências científicas sólidas e sejam implementadas de forma coordenada e eficaz pelas autoridades de saúde e pela população em geral. A conscientização da população sobre a importância das medidas de prevenção, como o uso de máscaras, o distanciamento social e a vacinação, também é essencial para o sucesso das estratégias de controle da pandemia.

Por fim, é importante ressaltar que a pandemia do COVID-19 trouxe à tona a necessidade de fortalecer os sistemas de saúde e investir em pesquisa e desenvolvimento de novas tecnologias e tratamentos para doenças infecciosas. A colaboração internacional também é fundamental para enfrentar pandemias globais como esta, e é necessário fortalecer a cooperação entre países e organizações para garantir a segurança e a saúde da população mundial.

1.3 Passos deste trabalho

Este trabalho tem como objetivo principal a modelagem discreta da dinâmica do COVID-19 ao longo do tempo, com base nos dados disponíveis sobre a doença. No próximo capítulo, será apresentada toda a metodologia utilizada, incluindo as definições de conceitos e algoritmos necessários para discretização do sistema. O objetivo é entender como a doença se espalha e evolui ao longo do tempo, permitindo prever cenários e identificar ações que possam minimizar seu impacto na população.

No capítulo seguinte, serão realizadas simulações baseadas na modelagem discreta desenvolvida, permitindo analisar diferentes cenários e estratégias de combate à doença. Além disso, serão apresentadas análises de sensibilidade para avaliar a robustez do modelo diante de diferentes hipóteses e cenários.

Por fim, o objetivo deste estudo é contribuir para o entendimento da dinâmica do COVID-19 e auxiliar na tomada de decisão de políticas públicas e medidas de saúde.

Espera-se que a modelagem discreta desenvolvida possa ser útil para prever a evolução da doença em diferentes contextos e subsidiar estratégias de combate mais eficazes e direcionadas. O conhecimento gerado neste trabalho pode ser aplicado a outras doenças infecciosas e epidemiológicas, ampliando sua relevância e utilidade para a sociedade.

2 Definições e conceitos utilizados neste trabalho

2.1 Probabilidade e Probabilidade Condicional

Morgado et al. (1991) define *Teoria das Probabilidade* como o ramo da Matemática que cria, desenvolve e em geral pesquisa *modelos* que podem ser utilizados para estudar experimentos ou fenômenos aleatórios. O modelo matemático utilizado para estudar um fenômeno aleatório particular varia em sua complexidade matemática, dependendo do fenômeno estudado.

Seja Ω um conjunto não vazio, denotado por *espaço amostral*. Comumente, Ω é referido como o conjunto de todos os resultados de um experimento aleatório (MAGALHÃES, 2006). A partir de Ω define-se uma família $\mathcal{A}$ de seus subconjuntos satisfazendo:

(i) $\varnothing \in \mathcal{A}$.

(ii) Se $A \in \mathcal{A}$ então $A^c \in \mathcal{A}$, em que $A^c = \Omega - A = \{\omega \in \Omega : \omega \notin A\}$.

(iii) $(A_n)_{n\geq 1}$ é uma sequência de elementos de $\mathcal{A}$, então

$$\bigcup_{n=1}^{\infty} A_n \in \mathcal{A}.$$

A família $\mathcal{A}$ é chamada uma σ-álgebra sobre Ω se seus elementos são denominados eventos de Ω.

Agora, define-se probabilidade como sendo uma função $P(\cdot)$ que atribui valores numéricos aos eventos do espaço amostral (Ω), satisfazendo as seguintes condições:

(i) $0 \leq P(A) \leq 1$, para todo evento $A \in \mathcal{A}$;

(ii) Se $A \subset B$, então $P(A) \leq P(B)$;

(iii) Se $(A_j)_{j\geq 1}$ é uma sequência de eventos do espaço amostral, dois a dois disjuntos, então

$$P\left(\bigcup_{j=1}^{\infty} A_j\right) = \sum_{j=1}^{\infty}(A_j).$$

A tripla ordenada $(\Omega, \mathcal{A}, P)$ definida pelos elementos acima é chamada *espaço de probabilidade*.

Em muitas situações práticas, o fenômeno aleatório com o qual é trabalhado pode ser separado em etapas. A informação do que ocorreu em uma determinada etapa pode influenciar nas probabilidades de ocorrências das etapas sucessivas. Nestes casos, houve ganho de informação e deve-se "recalcular" as informações de interesse, essas probabilidades "recalculadas" são chamadas de probabilidades condicionais, e são definidas da seguinte maneira: dado dois eventos A e B, a probabilidade condicional de A dado que ocorreu B é representada por $P(A|B)$ e é definida por:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}, \quad P(B) > 0.$$

Caso $P(B) = 0$, $P(A|B)$ pode ser definido arbitrariamente; neste contexto será utilizado $P(A|B) = P(A)$

2.1.1 Variáveis Aleatórias

De maneira sucinta, uma variável aleatória é um característico numérico de uma experimento aleatório, ou seja, é uma função que associa um elemento de um espaço amostral a um número de uma forma como se segue: se $(\Omega, \mathcal{A}, P)$ é um espaço de probabilidade, denote por $X : \Omega \to \mathbb{R}$, que toma um elemento $\omega \in \Omega$ e associa a um número real $X(\omega) \in \mathbb{R}$, tal que

$$X^{-1}(-\infty, x) \in \mathcal{A},$$

para todo $x \in \mathbb{R}$. Nesse caso, X é denominada uma *variável aleatória*. Recomenda-se a leitura do livro (MAGALHÃES, 2006) para mais detalhes.

Diz-se que uma variável aleatória X é *discreta* se $X \in \{x_1, x_2, \ldots\}$ e sua *função de probabilidade* $p(x_i) = P(X = x_i)$ é definida por:

$$p(x_i) = P(\{\omega \in \Omega \mid X(\omega) = x_i\}).$$

e, sua *função de distribuição acumulada* é dada por:

$$F_X(x) := P(X \leq x) = \sum_{x_i \leq x} p(x_i),$$

para todo $x \in \mathbb{R}$.

Agora, diz-se que uma variável aleatória X é contínua se existe uma função $f = f_X$, denominada *função densidade de probabilidade* tal que

$$F_X(x) := \int_{-\infty}^{x} f_X(t)\, dt.$$

Neste caso, F_X também é a função de distribuição acumulada de X.

Para ambos os casos, tem-se as seguinte propriedades:

Variáveis Discretas		Variáveis contínuas	
1)	$p(x_i) \geq 0,\ \forall x_i \in \mathbb{R}$	1)	$f_X(x) \geq 0,\ \forall x \in \mathbb{R}$
2)	$\sum_i p(x_i) = 1$	2)	$\int_{\mathbb{R}} f_X(x)dx = 1$

É fácil ver que, em todos os casos:

F1. $0 \leq F_X(x) \leq 1$, para todo $x \in \mathbb{R}$.

F2. F_X é uma função crescente, ou seja, para todo par x e y com $x < y$, tem-se $F_X(x) \leq F_X(y)$.

F3. F_X é uma função contínua à direita, ou seja, para todo $x_0 \in \mathbb{R}$,

$$\lim_{x \to x_0^+} F_X(x) = F_X(x_0).$$

2.1.2 Esperança e Variância

De maneira objetiva,

Definição 2.1.1. Sejam X uma variável aleatória discreta com $X \in \{x_1, x_2, \ldots\}$ com função de probabilidade $p(x)$ e Y uma variável aleatória contínua com função densidade de probabilidade $f_Y(y)$. Então,

$$\mathbb{E}(X) = \sum_i x_i p(x_i) \qquad\qquad \mathbb{E}(Y) = \int_{-\infty}^{\infty} y f_Y(y)\, dy \tag{2.1}$$

$\mathbb{E}$ é denominada a *esperança* das variáveis associadas e, obviamente, no caso contínuo está definida se a integral existir.

Da mesma forma,

Definição 2.1.2. Seja X uma variável aleatória qualquer. No caso em que das esperanças envolvidas estiverem bem definidas:

$$\text{Var}(X) = \mathbb{E}(X - \mu)^2 = \mathbb{E}(X^2) - \mu^2, \tag{2.2}$$

em que $\mu_X = \mathbb{E}(X)$. A função $\text{Var}(X)$ é denominada a *variância* de X.

A esperança de uma variável aleatória é a principal medida de posição desta variável, enquanto a variância é a principal medida de dispersão. Toda variável aleatória é bem entendida a partir destas duas medidas.

2.2 Processos Estocásticos

Em Ibe (2013) um processo estocástico é definido como um coleção indexada de variáveis aleatórias $\{X(t); t \in T\}$, onde T normalmente é composto por números não-negativos, e $X(t) = X_t$ é a característica mensurável de interesse no tempo t. Os valores assumidos por X_t são chamados de estados, e o conjunto de todos os possíveis valores forma o espaço de estado do processo. Pode-se classificar os processos estocásticos analisando-se:

(i) **O espaço de estados**, se X for um conjunto de estados enumeráveis, ou seja, X assume uma quantidade enumerável de valores, $X(t)$ é um "processo de estados discretos" ou, como é usualmente referido, uma "cadeia", caso contrário o processo é designado por "processo em estado contínuo";

(ii) **A variável temporal**: se T for enumerável, $X(t)$ é um "processo de tempo discretos", caso contrário $X(t)$ é designado por "processo em tempo contínuo";

(iii) **Características estatísticas das variáveis aleatórias**: o processo diz-se *estacionário* se o seu comportamento for independente do tempo, também é chamado *markoviano* se for estacionário e gozar da propriedade de Markov ou da "perda de memória", em que seu comportamento futuro depender apenas do estado presente, ou ainda, Semi-Markov em que acontecimentos sucessivos deixam de estar "restritos" à distribuição exponencial, podendo seguir qualquer distribuição de probabilidade

2.3 Processos Markovianos

Processos de Markov constituem um tipo especial de processo estocástico, que segundo Ibe (2013) possui a propriedade de que as probabilidades associadas com o

processo num dado instante do futuro dependem somente do estado presente, sendo, portanto, independentes dos eventos no passado. Desse modo, os processos markovianos são caracterizados pelo que se designa como “falta de memória”, ou mais formalmente, o processo estocástico $X(t); t \in T$ é dito markoviano, quando para qualquer tempo, $t_0 < t_1 < ... < t_n < t$, a probabilidade condicional de X_t para os valores dados de $X_{t_0}, X_{t_1}, ..., X_{t_n}$ dependem somente de X_{t_n}.

Ainda segundo Ibe (2013) é fundamental no estudo de processo de Markov a noção de estado. Propriedade comum entre indivíduos (ou objetos) caracterizam o que designamos por estados. Os processos de Markov sempre envolvem a a variável “tempo”, seja considerada de forma discreta onde o tempo varia em saltos, ou de formas contínuas podendo assumir valores reais.

2.4 Diagrama de fases

O diagrama de fases é uma representação gráfica dos estados físicos ou fisicoquímicos de um sistema sob diferentes condições de temperatura e pressão. Um diagrama de fase típico tem pressão no eixo das ordenadas e temperatura no eixo das abscissas.

Gráficos ternários ou diagramas ternários são representações triangulares das composições de objetos que podem ser expressas por misturas de três componentes. Os gráficos ternários são encontrados em diversas áreas, como a química e geologia.Na geologia comumente são utilizados para demonstrar a composição de rochas e minerais, dependendo do conteúdo mineral, proporção de extremidades ou mesmo composição química. No presente trabalho, serão utilizados os diagramas ternários, tais como exemplificado na Figura 1 abaixo.

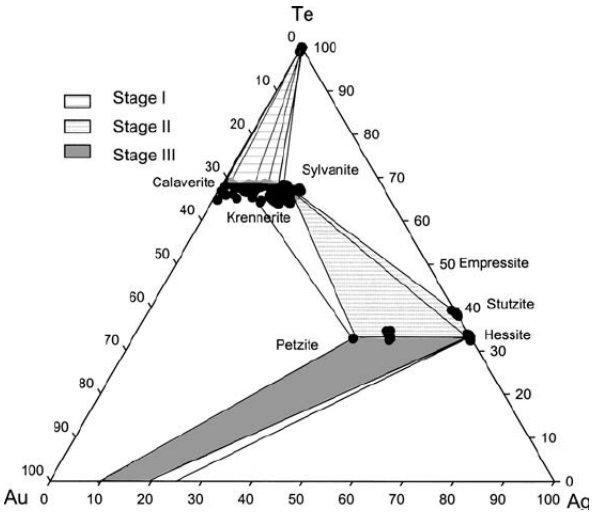


Figura 1 – Exemplo de um sistema Au, Ag e Te representado por um digrama ternário. Fonte: Pals e Spry (2003).

No trabalho Pals e Spry (2003), apresenta-se o digrama da Figura 1 constituído por três elementos, ouro (Au), prata (Ag) e telúrio (Te) em proporções atômicas para os minerais analisados naquele estudo do depósito de outro epitermal do Imperador, em Vatukoula, Fiji. As linhas sólidas indicam composições representativas de fases coexistentes. As assembleias de estágio II são ricas em Te e Au, enquanto as assembleias de estágio III são ricas em Ag. Os conjuntos de metais preciosos do estágio IV são pobres em Te e contêm teluretos ricos em Ag mais ouro nativo, denominado *electrum*. Observe o continuum de composições entre calaverita, krennerita e silvanita.

Os diagramas ternários de fase como ilustrados nessa Seção são inspirações para o trabalho ora desenvolvido. O que se verá é a definição do espaço de possíveis estados de um sistema de interesse de maneira natural como um sistema ternário.

2.5 Grafos

As definições desta Seção podem ser encontradas em diversos livros tais como Bondy, Murty et al. (1976), Bollobás (1998), e Diestel (2003). Os grafos serão utilizados nesse trabalho uma representação discreta dos sistemas ternários como definido na Seção 2.4.

2.5.1 Definições

Definição 2.5.1 (Grafo)**.** É denominado como *grafo* um par $G = (V, E)$ que satisfaz $E \subseteq [V]^2$, onde $[V]^2$ representa o conjunto de todos os subconjuntos de 2 elementos de V.

Para evitar ambiguidades de notação, assume-se tacitamente que $V \cap E = \varnothing$. Os elementos de V são os vértices do grafo G e os elementos de E são as suas arestas. O conjunto de vértices de um grafo G é referido como $V(G)$ e a suas arestas definidas como $E(G)$. Uma aresta $\{x, y\}$ é geralmente escrita como xy. Os grafos são usados para representar relações e conexões em diversos campos, incluindo ciência da computação, matemática, física, biologia, entre outros. Existem diversos tipos de grafos, cada um com suas próprias propriedades e aplicações.

É possível se fazer uma apresentação gráfica de um grafo representando vértices por pontos no espaço e as arestas como linhas unindo dois vértices. Por exemplos, o Grafo de Petersen[1] que aparece na Figura 2, é um grafo com 10 vértices e 15 arestas. Ele é conhecido por não ser bipartido, ou seja, não é possível dividir seus vértices em dois conjuntos ditos independentes, de modo que todas as arestas tenham extremos estejam em conjuntos distintos. Outros exemplos de grafos incluem o grafo completo, grafo cujos vértices estão todos conectados entre si; o grafo bipartido, para o qual o conjunto de vértices pode ser dividido em dois conjuntos independentes; e uma árvore, que é um grafo para o qual todos os vértices estão conectados por um única sequência vértice-aresta, chamada caminho (Veja Figuras 3 (a)-(c)).

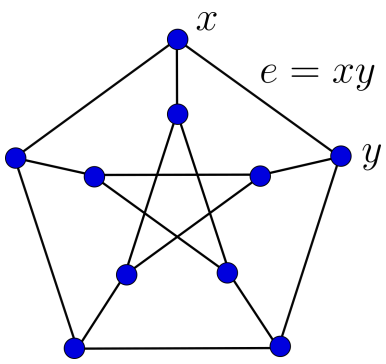
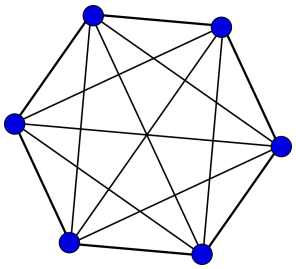


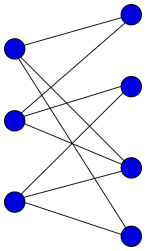
Figura 2 – Grafo de Petersen. Estão em destaque dois vértices x e y e uma aresta $e = xy$. Fonte: Autora.

Definição 2.5.2 (Tamanho de um Grafo)**.** Seja $G = (V, E)$ um grafo. A *ordem* de um grafo é o número de vértices em G, isto é, $|V|$. O *tamanho* de G é o número de arestas, isto é, $|E(G)|$.

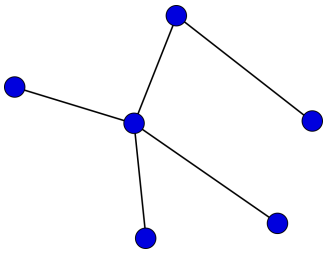
[1] Devido a Julius Petersen (1839-1910).



(a) Grafo completo com 6 vértices.



(b) Grafo bipartido.



(c) Árvore.

Figura 3 – Representação de três grafos. Em (a) o grafo é denominado C_6. De modo geral, C_k é um grafo completo com k vértices. Em (b), os vértices da esquerda estão ligados somente a vértices da direira e vice versa. Em (c), para cada par de vértices existe e é único *caminho* ligando esses vértices. Fonte: Autora.

Definição 2.5.3 (Distância de dois vértices)**.** A *distância* $d_G(u, v)$ em G de dois vértices u, v é o comprimento do caminho mais curto uv em G, se não existe tal caminho, definimos $d(u, v) := \infty$. A maior distância entre quaisquer dois vértices no grafo G é o *diâmetro* de G, indicados por $diam(G)$.

Definição 2.5.4 (Adjacência)**.** Dado um *grafo* G, dois vértices x, y de G são *adjacentes*, ou *vizinhos*, se xy é uma aresta de G.

Na definição utilizada não será permitida a existência de laços (arestas cujos vértices iniciais e finais são iguais), nem multi-arestas (duas ou mais arestas com os mesmos vértices iniciais e finais).

Definição 2.5.5 (Grau de um vértice)**.** Seja um grafo $G = (V, E)$ com u e $v \in V$. O grau de v é o número de arestas com as quais v é incidente. O grau de v se denota por $\Gamma_G(v)$ ou, se não houver risco de confusão, simplesmente por $\Gamma(v)$. Em outras palavras $\Gamma(v) = |N(v)|$.

Na aplicação de grafos para modelagem, o grau de um vértice representa as relações entre indivíduos. Nesse caso essas ligações estão relacionadas com quais relações estamos querendo desenvolver para esses indivíduos.

Definição 2.5.6 (Vizinhança do vértice v_ℓ)**.** Considerando o grafo $G = (V, E)$ e supondo que u e v sejam vértices de G. Se u e v são adjacentes, dizemos que u e v são vizinhos. O conjunto de todos os vizinhos de um vértice v é chamado vizinhança de v e se denota por $Nb(v)$, isto é, $Nb(v) = \{u \in V : uv \in \mathbb{E}(G)\}$.

Para esse trabalho é importante salientar que os sistemas trabalhados são sistemas dinâmicos complexos com relações locais, ou seja, um vértice muito distante do vértice atual não irá afetar a dinâmica naquele instante, pois a probabilidade de que um vértice passe para outro em um período muito curto é de zero, consequentemente a dinâmica se dá entre vizinhanças. Além disso, é importante destacar que o grafo proposto apresenta um laço em cada vértice, o que garante a existência de uma probabilidade não nula de permanência no mesmo estado.

2.5.2 Representação de um Grafo

Um grafo pode ser representado através da *lista de adjacência* onde é armazenado o relacionamento entre os vértices de uma estrutura de listas e também pode ser representado por matrizes. Uma representação importante é através da matriz de adjacência.

Definição 2.5.7 (Matriz de Adjacência)**.** Dado um *grafo* G, sua *matriz de adjacência* $M_{m\times m}$ (onde m é o total de vértices do grafo) é denominada por $A = (a_{ij})_{n\times n}$ e definida da seguinte forma:

$$a_{ij} = n, \quad \text{se existem } n \text{ arestas ligando } v_i \text{ a } v_j$$

Considerem o grafo da Figura 4.

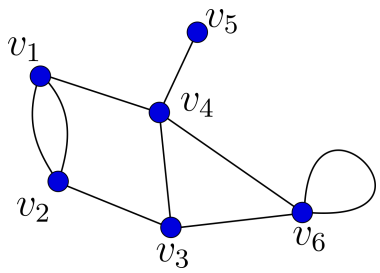


Figura 4 – Exemplo de grafo. Fonte: Autora

A matriz de adjacência do grafo acima é dado por:

$$A = \begin{array}{c} \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{array} \begin{array}{c} \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\ \begin{pmatrix} 0 & 2 & 0 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \end{array}$$

Note que para entre v_1 e v_2 existem duas arestas ligando esses vértices. Neste caso, diz-se que são arestas em paralelo ou que entre v_1 e v_2 existem múltiplas arestas. Já para o vértice v_6 existe uma aresta que liga ele a ele próprio. Nestes casos, diz-se que tal aresta é um laço.

Definição 2.5.8 (Matriz de Incidência)**.** A matriz de incidência $B = (b_{ij})_{n\times m}$ de um grafo $G = (V, E)$ com $V = \{v_1, v_2, \ldots, v_n\}$ e $E = \{e_1, e_2, \ldots, e_m\}$ é definida por

$$b_{ij} = \begin{cases} 1, & \text{se } e_j \text{ é incidente ao vértice } v_i, \\ 0, & \text{caso contrário.} \end{cases}$$

Os grafos são amplamente utilizados em diversas áreas do conhecimento, desde a análise de redes sociais na internet até a otimização de processos produtivos em grandes indústrias. Neste trabalho, o uso dos grafos se dará como uma ferramenta de modelagem de um sistema dinâmico, permitindo uma análise mais precisa das interações entre os elementos que compõem o sistema.

2.6 Percolação

A Teoria da Percolação foi proposta pelos matemáticos Broadbent e Hammersly em 1957 (ESSAM, 1980) para estudar a propagação de fluidos em meios desordenados. É bastante relevante em física para explicar eventos como recuperação de petróleo a partir de meios porosos, fogos florestais, modelos de epidemia, redes e terremotos (SAHIMI, 2023). O modelo de percolação, permite explicar este tipo de eventos e introduzir alguns conceitos fundamentais dos sistema complexos (grafos), a Teoria da Percolação é um ramo da Teoria das Probabilidades que trata das propriedades dos meios aleatórios agregadas a conceitos geométricos e probabilísticos.

Existem dois tipos de percolação (STAUFFER; AHARONY, 2018): percolação por ligações e percolação por pontos. Considerando um *cluster* qualquer, a percolação por ligações é o estudo de fenômenos de percolação nas arestas, o modelo da infiltração de óleo em solos permeáveis, enquanto que a percolação por pontos estuda os fenômenos de percolação nos vértices (nós) sendo o caso da propagação de fogos florestais. Estes dois tipos de percolação, embora diferentes, apresentam propriedades semelhantes que são estudadas sobre um espaço L^d, sendo d a dimensão.

Um processo de percolação consiste na propagação do estado de uma célula ativa às células vizinhas que, depois de ativadas, continuam o processo de propagação. O processo

termina quando não há mais células do agregado que possam ser ativadas. A duração do processo de percolação depende de dois fatores, sendo eles o tamanho do agregado e a forma como estão ligados. É evidente que quanto maior for o agregado maior será a duração do processo de percolação e um agregado muito ligado percola muito mais rapidamente do que um agregado com poucas ligações.

Nesta seção, foram apresentadas as principais ferramentas que serão utilizadas ao longo deste trabalho. Nas próximas seções, serão apresentadas aplicações específicas dessas ferramentas, incluindo a modelagem do comportamento da doença dada a aplicação proposta com o uso de grafos.

3 Objetivos

3.1 Objetivo Geral

Descrever para um grafo especialmente construído uma dinâmica ao longo do tempo, que represente a mudança de estados, que são descritos pelos vértices desse grafo. O grafo é definido por um algoritmo proposto e que deve ser capaz de responder com precisão fixada, determinada pelo usuário, às questões tais como em qual estado se encontra o sistema após determinado tempo a partir de um estado inicial, ou quantas mudanças de estados são necessárias para que o sistema adquira determinada configuração.

3.2 Objetivos Específicos

- Descrever um sistema que pode ter três estados puros ou combinações de estados.
- Gerar um sistema de coordenadas que localiza o estado atual do sistema relativamente aos três estados puros.
- Descrever a dinâmica estocástica (modelo de difusão) dos estados de um sistema específico dada por uma função de distribuição acumulada que descreva a probabilidade de um estado do sistema a ser alcançado em um determinado intervalo de tempo a partir de um ponto inicial arbitrário.
- Escrever o algoritmo que discretize o espaço de estados, gerando de maneira sistemática um grafo e o modelo de difusão sobre o mesmo.

4 Metodologia

Como descrito na Capítulo 1, esse trabalho visa buscar um método de solução de dinâmicas de estados utilizando estruturas discretas. As estruturas discretas utilizadas serão os grafos, que surgirão naturalmente após aplicação do algoritmo a ser construído para esse fim.

4.1 O Espaço de Estados

Seja $\mathcal{T}$ um sistema constituído por três estados puros distintos e únicos, denotados por S_0, S_1 e S_2, ou como uma combinação convexa destes três estados:

$$x_0S_0 + x_1S_1 + x_2S_2 \tag{4.1}$$

em que $x_j \geq 0$, $j = 0, 1, 2$ e $x_0 + x_1 + x_2 = 1$. Por exemplo, se for considerada uma determinada população e relativamente a alguma doença, cada indivíduo desta população pode estar suscetível à doença, p. ex. S_O, ele pode estar infectado pelo mal, p. ex. S_1, ou estar recuperado, ou seja, curado da doença, S_2, nesse contexto. Logo, denota-se o *estado da população* (melhor que sistema, neste caso) à quantidade de pessoas nos estados suscetível, infectado, ou recuperado. Usualmente, indica-se para cada caso o percentual de cada subgrupo em relação ao total da população. Deve-se notar que se for denotado por $S_\mathcal{T}$ o estado atual do sistema o mesmo pode ser descrito pelas coordenadas (x_0, x_1, x_2) relativamente à base de estados puros $S_0 = (1, 0, 0)$, $S_1 = (0, 1, 0)$ e $S_2 = (0, 0, 1)$. O *espaço de estados*, o conjunto que representa todos os estados possíveis do sistema $\mathcal{T}$ pode ser representado como um triângulo equilátero (Figura 5) e as coordenadas do estado atual ser dada via coordenadas baricêntricas relativas aos estados puros. Denote-se o espaço de estados por $\mathbb{T}$ e, obviamente é uma variedade de dimensão 2 com bordo, representada em $\mathbb{R}^3$ pelo conjunto:

$$\{(x, y, z) \in \mathbb{R}^3 \,:\, x + y + z = 1, x \geq 0, y \geq 0, z \geq 0\}.$$

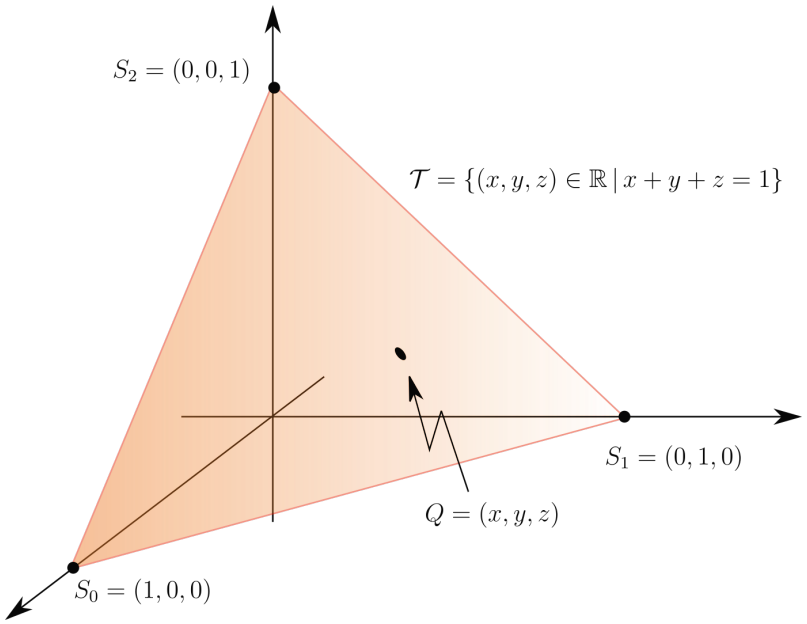


Figura 5 – Espaço de estados $\mathcal{T}$. Fonte: Autora

O estado atual é considerado como uma função do tempo, $S_\mathcal{T} = S_\mathcal{T}(t)$ e a dinâmica de tal estado é dada pela função de probabilidade condicional a seguir:

$$\pi(\tau, E \,|\, \sigma, \mathbf{x}) \tag{4.2}$$

é a probabilidade de se ter o estado $S_\mathcal{T}(\tau) \in E \subset \mathcal{T}$ (considerando E como um conjunto arbitrário, aberto e mensurável), no instante τ, tal que no instante σ o estado era $S_\mathcal{U}(\sigma) = \mathbf{x} = (x_0, x_1, x_2)$.

Uma possível modelagem para a dinâmica acima é considerando a dinâmica como um processo de difusão (BEAL; BOHLEN et al., 1957; PORTENKO, 1990; JACOBS, 1935). Para tal, define-se, primeiramente, um conjunto de variáveis indexadas pelo tempo, chamado *processo de Markov*. Seja $S_t(\omega) = S(\omega, t)$ um processo estocástico que representa a posição do estado de um sistema dado um movimento aleatório. A completa caracterização desse movimento aleatório é dada pela densidade de probabilidade condicional

$$\pi(\tau, \mathbf{y} \,|\, \sigma, \mathbf{x}) = P(S\tau = \mathbf{y} \,|\, S_\sigma = \mathbf{x}), \tag{4.3}$$

e relacionada à distribuição condicional $\Pi(\tau, E \,|\, \sigma, \mathbf{x})$, isto é,

$$\Pi(\tau, E \,|\, \sigma, \mathbf{x}) = \int_{E \subseteq \mathcal{T}} \pi(\tau, \mathbf{y} \,|\, \sigma, \mathbf{x}) d\mathbf{y}$$

com

$$\int_\mathcal{T} \pi(\tau, \mathbf{y} \,|\, \sigma, \mathbf{x}) d\mathbf{y} = 1 \tag{4.4}$$

Para um processo de difusão é requerida que seja válida a *igualdade de Markov*

$$\pi(\tau, \mathbf{y} \,|\, \sigma, \mathbf{x}) = \int_{\mathbb{T}} \pi(s, \mathbf{z} \,|\, \sigma, \mathbf{x}) \cdot \pi(\tau, \mathbf{y} \,|\, s, \mathbf{z}) d\mathbf{z} \tag{4.5}$$

Considera-se, também, que seja válida a condição de *continuidade forte* para o processo de Markov, isto é,

$$\lim_{\Delta\sigma \to 0} \frac{1}{\Delta\sigma} \int_{|\mathbf{y}-\mathbf{x}|>\delta} \pi(\sigma, \mathbf{y} \,|\, \sigma - \Delta\sigma, \mathbf{x}) d\mathbf{y} = 0 \tag{4.6}$$

para todo $\delta > 0$. Assim a probabilidade de que um ponto aleatório tomará grandes incrementos em um pequeno intervalo de tempo tem medida nula.

E, ainda, serão consideradas as seguintes condições:

(a) As funções

$$\frac{\partial\, \pi(\tau, \mathbf{y} \,|\, \sigma, \mathbf{x})}{\partial\, x^i} \quad \text{e} \quad \frac{\partial^2\, \pi(\tau, \mathbf{y} \,|\, \sigma, \mathbf{x})}{\partial\, x^i\, x^j},$$

existem e são funções contínuas para todo $i, j \in \{0, 1, 2\}$, para todo $\mathbf{x}, \mathbf{y} \in \mathbb{T}$ e para todo $\sigma, \tau \in \mathbb{R}^+$, $\sigma < \tau$.

(b) Para todo $\delta > 0$ e $i, j = 0, 1, 2$, os limites abaixo existem:

$$\lim_{\Delta\sigma \to 0} \frac{1}{\Delta\sigma} \int_{|\mathbf{x}-\mathbf{y}|<\delta} (y^i - x^i) \pi(\tau, \mathbf{y} \,|\, \sigma - \Delta\sigma, \mathbf{x}) d\mathbf{y} = b^i(\sigma, \mathbf{x}), \tag{4.7}$$

$$\lim_{\Delta\sigma \to 0} \frac{1}{\Delta\sigma} \int_{|\mathbf{x}-\mathbf{y}|<\delta} (y^i - x^i)(y^j - x^j) \pi(\tau, \mathbf{y} \,|\, \sigma - \Delta\sigma, \mathbf{x}) d\mathbf{y} = 2a^{ij}(\sigma, \mathbf{x}). \tag{4.8}$$

Estes dois limites são costumeiramente considerados uniformes no sentido da definição segundo ε's e δ's. Também pelo Teorema de Heine (KRANTZ, 2005), quando o processo está restrito a um conjunto compacto do $\mathbb{R}^n$ a uniformidade é automática.

Teorema 4.1.1 (Kolmogorov)**.** *Sobre as condições acima,* $\pi(\tau, \mathbf{y} \,|\, \sigma, \mathbf{x})$ *como uma função de* σ *e* $\mathbf{x}$ *satisfaz:*

$$a^{ij}(\sigma, \mathbf{x}) \frac{\partial^2 \pi}{\partial\, x^i \partial\, x^j} + b^i(\sigma, \mathbf{x}) \frac{\partial\, \pi}{\partial\, x^i} = -\frac{\partial\, \pi}{\partial\, \sigma} \tag{4.9}$$

A demonstração do Teorema 4.1.1 pode ser encontrado em (ANTONELLI; STROBECK, 1977).

Quando a^{ij} e b^i não dependem explicitamente do tempo, dizemos que a difusão markoviana é *homogênea no tempo*. Neste caso, com $\tau > \sigma$:

$$\pi(\tau, \mathbf{y} \,|\, \sigma, \mathbf{x}) = \pi(\tau - \sigma, \mathbf{y} \,|\, 0, \mathbf{x}), \tag{4.10}$$

Nessa Seção formam apresentadas algumas das propriedades e resultados que serão usados para a definição da dinâmica no caso proposito por este trabalho. Para isso, será primeiramente construído o espaço no qual se dá a dinâmica.

4.2 Construção do grafo de estados puros e mistos

Este trabalho tem por objetivo o estudo de uma dinâmica a ser definida sobre um grafo especial, construído como representação do estaço de estados $\mathcal{T}$. Nesta Seção, será explicitada esta construção.

Seja S um sistema contendo três estados puros, no qual denominaremos por S_0, S_1, S_2 onde o espaço desse sistema é representado por um triângulo equilátero com cada vértice representando um desses estados, anteriormente denominado por $\mathcal{T}$. O vértice S_0 é oposto a base do triângulo que é determinada pelos vértices S_1 e S_2, os vértices são nomeados a partir do vértice S_0 seguindo-se em sentido horário. A Figura 6 representa esse sistema inicial.

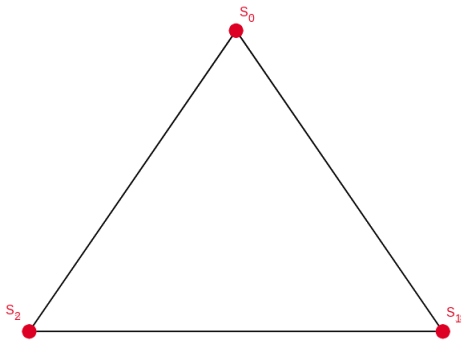


Figura 6 – Representação do sistema S e seus três estados puros. Fonte: esta pesquisa.

Na construção do *grafo de estados mistos*, denominado de $\mathcal{G}_N$, é necessário a estruturação de linhas, as quais chamamos de *linhas de isoestados* relativas aos estados puros. Essas linhas são paralelas entre si e opostas ao estado puro referenciado e dividirão os lados adjacentes a este estado em N pedaços iguais (conforme escolha do modelador). São denominadas isoestados, pois têm a mesma coordenada relativa ao respectivo estado puro. Nas Figuras 7, 8 e 9 estão representadas as isolinhas relativas aos estados puros S_0, S_1 e S_2 respectivamente, com o $N = 4$.

Como dito, a quantidade de isolinhas, N, será definida pelo usuário e será a mesma para todos os estados puros. Quando traçadas as isolinhas dos três estados puros na mesma representação é gerada uma malha triangular sobre o sistema $\mathcal{T}$, como mostra a Figura 10.

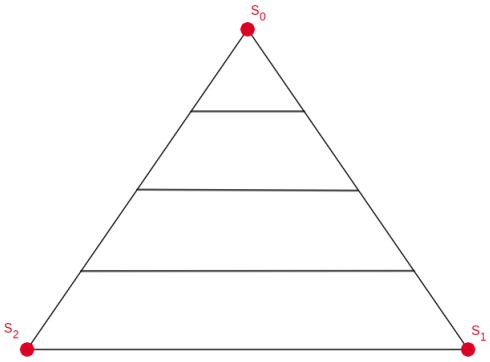


Figura 7 – Isolinhas relativas ao estado puro S_0. Fonte: esta pesquisa.

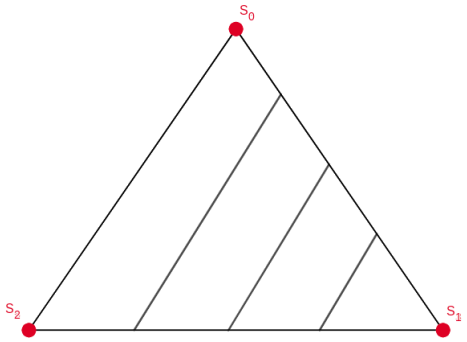


Figura 8 – Isolinhas relativas ao estado puro S_1. Fonte: Autora.

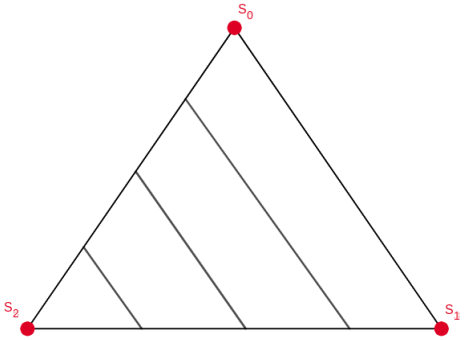


Figura 9 – Isolinhas relativas ao estado puro S_2. Fonte: esta pesquisa.

Dada essa malha triangular, tomam-se os baricentros de cada um dos triângulos gerados como sendo os vértices do grafo $\mathcal{G}_N$ e cada vértice v_ℓ estará ligado ao vértice v_i se os triângulos possuem um lado em comum. Tomar-se o baricentro de cada triângulo é coerente com a média local de cada processo que será modelado, lembrando-se que a média é justamente o *centro de massa* da variáveis que se está trabalhando.

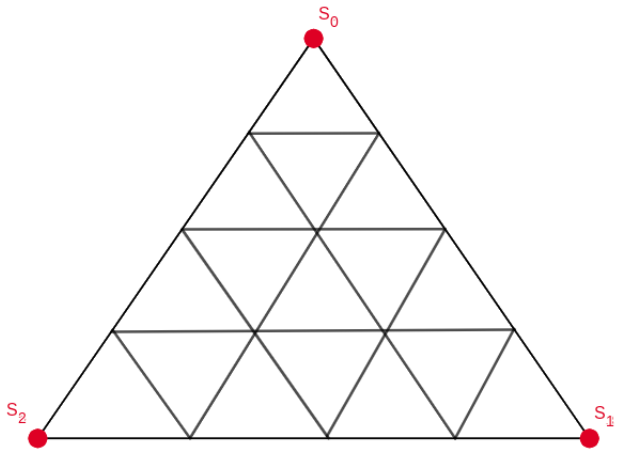


Figura 10 – Representação do sistema S e seus três estados puros. Fonte: esta pesquisa.

No caso de um triângulo equilátero, todos os pontos notáveis (baricentro, ortocentro, incentro e circuncentro) recaem no mesmo ponto. Na Figura 11 pode-se verificar a representação dos baricentros gerados.

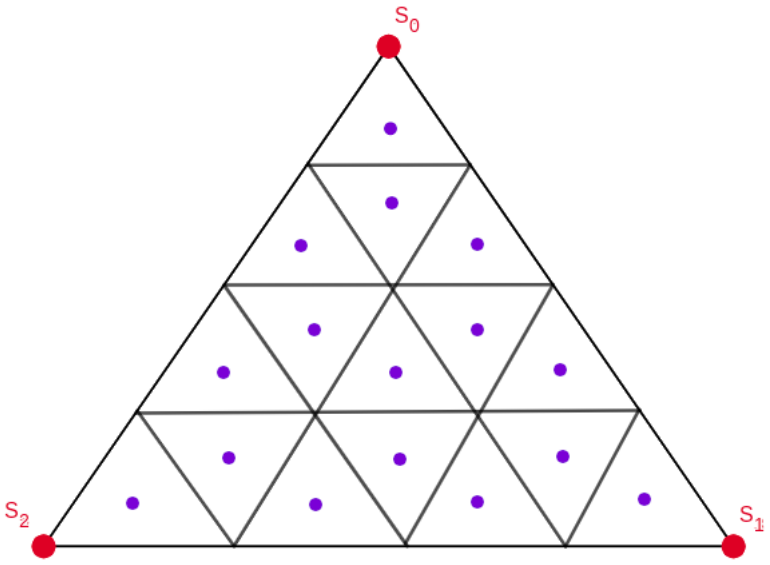


Figura 11 – Baricentros dos triângulos formados pelas *isolinhas*. Fonte: esta pesquisa.

Dado o grafo gerado pelo procedimento descrito acima, pode-se identificar seus vértices e, neste caso, eles serão denominados por v_ℓ onde $\ell = 1, 2, \ldots, N^2$. O vértice v_1 será o primeiro vértice localizado logo abaixo do vértice S_0 e os vértices seguintes serão numerados da esquerda pra direta e de cima para baixo (na representação do grafo como na Figura 12) como na e em ordem crescente. Os vértices colineares são os pontos que pertencem a mesma reta (linha), neste caso, paralela ao lado $S_1 - S_2$, essas

linhas denominamos como *níveis* e serão identificados por $L_1(S_0), L_2(S_0), L_3(S_0)$ e assim sucessivamente. Na Figura 12 pode-se verificar os vértices e seus respectivos níveis.

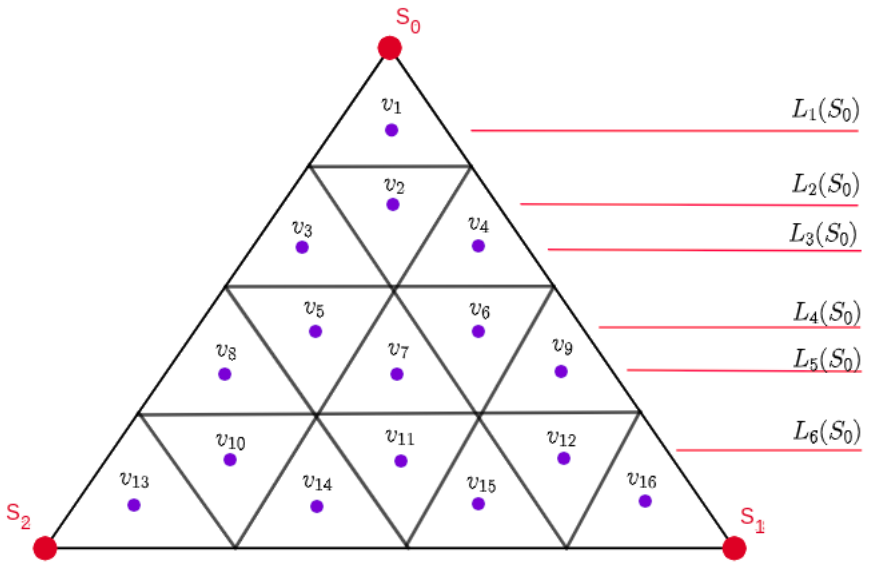


Figura 12 – Representação do sistema S com o grafo pronto sobre ele. Fonte: esta pesquisa.

Alguns resultados para os grafos de estados são apresentados a seguir.

Proposição 4.2.1 (Quantidade de níveis)**.** *Dado um grafo $\mathcal{G}_N$ que representa o espaço de um sistema S contendo três estados puros, tem a quantidade de níveis dado por $(2N-1)$, onde o N é a quantidade de partes em que as arestas do triângulo serão divididas, posteriormente definiremos como* quantil.

Demonstração. Pela lei de formação dos grafos, vemos que a cada nova sub-divisão, ou acréscimo de faixa, acrescentamos dois novos níveis ao grafo, pois excetuando-se a primeira faixa (que contém apenas um nível), todas as faixas contém dois níveis, como tem-se tantas faixas quanto seja o n, tem-se que o número de níveis é dado por $N = (2n-1)$. □

Proposição 4.2.2 (Diâmetro do Grafo)**.** *Todo grafo $\mathcal{G}_N$ que representa o espaço de estados de um sistema S que contém três estados puros, tem o seu diâmetro dado por $N = (2n-2)$, onde n é a quantidade de partes em que as arestas do triângulo são divididas.*

Demonstração. Por resultados da Geometria Euclidiana, tem-se que em um triângulo equilátero a maior distância entre dois pontos que pode existir é a distância entre dois de seus vértices. Assim dada a definição de $diam(\mathcal{G}_N)$, tem-se que ele será dado pela distância

entre os vértices do grafo, mais próximos dos vértices do triângulo, como o triângulo é equilátero, pode-se tomar a última faixa como parâmetro para definir o diâmetro do grafo. Como na última faixa tem-se $2n-1$ vértices, em linha, e precisamos de 2 vértices para obter uma aresta, então tem-se $2n-1-1 = 2n-2$ arestas que é o $diam(\mathcal{G}_N)$. □

As proposições só são válidas a partir de $n = 2$, pois para $n = 1$, existe apenas um vértice cujo grau é $|\Gamma(v_1)| = 1$.

Proposição 4.2.3 (Quantidade de vértices com grau 1)**.** *Dado um grafo $\mathcal{G}_N$ que representa o espaço de um sistema S contendo três estados puros, a quantidade de vértices com grau 1 é definido por $|\Gamma(v_1)| = 3$, em que N é a quantidade de partes na qual as arestas do triângulo $\mathbb{T}$ serão divididas.*

Proposição 4.2.4 (Quantidade de vértices com grau 2)**.** *Dado um grafo $\mathcal{G}_N$ que representa o espaço de um sistema S contendo três estados puros, a quantidade de vértices com grau 2 é definido por $|\Gamma(v_2)| = 3(N-2)$, onde o N é a quantidade de partes em que as arestas do triângulo serão divididas.*

Proposição 4.2.5 (Quantidade de vértices com grau 3)**.** *Dado um grafo $\mathcal{G}_N$ que representa o espaço de um sistema S contendo três estados puros, a quantidade de vértices com grau 2 é definido por $|\Gamma(v_3)| = N^2 - 3N + 3$, onde o N é a quantidade de partes em que as arestas do triângulo serão divididas.*

A demonstração das Proposições 4.2.3 – 4.2.5 podem ser feitas facilmente por um simples processo de contagem.

4.3 Rotulando os vértices de um grafo de estados mistos

Nesta Seção, será apresentada uma forma bem definida de rotulação dos vértices do grado $\mathcal{G}_N$, de modo a se sistematizar uma dinâmica sobre este grafo.

Dado um grafo $\mathcal{G}_N$, L_{v_ℓ} é o nível no qual o vértice se encontra, ou seja, qual a isolinha relativa ao estado S_0 o vértice se encontra no grafo. Já a posição p_{v_ℓ} do vértice v_ℓ no nível L_{v_ℓ} é contada da esquerda para direita na ordem dos vértices de $\mathcal{G}_N$.

Pela Proposição 4.2.1 sabe-se que o grafo $\mathcal{G}_N$ possui $2N-1$ níveis. Pode-se verificar também a quantidade de posições por níveis:

- Se nível do grafo $\mathcal{G}_N$ é par, digamos k, tem-se que a quantidade de posições é dada por $k/2$;
- Caso o nível k seja ímpar, a quantidade de posições por nível é dada por $(k+1)/2$.

Dessa forma, os níveis dados por $2m-1$ e $2m$ têm m vértices.

Seja m o índice do nível L_m, $m = 1, 2, \ldots, 2N-1$, onde N é o inverso do quantil escolhido inicialmente. Se m é par, os vizinhos (i, j, k) do vértice v_ℓ que está na posição i do nível m, que é representado por (m, i), é dado por:

$$Nb(m,i) = \Big\{(m-1,i),(m+1,i),(m+1,i+1)\Big\}, \quad i = 1, \ldots, \frac{m}{2}. \tag{4.11}$$

Se m é ímpar, a vizinhança do vértice que está na posição i do nível m é:

$$Nb(m,i) = \begin{cases} \{(m-1,1),(m+1,1)\}, & \text{se} \quad i = 1, \\ \{(m-1,i-1),(m-1,i),(m+1,i)\}, & \text{se} \quad i = 2, \ldots, \frac{m+1}{2}-1, \\ \left\{\left(m-1,\frac{m+1}{2}\right),\left(m+1,\frac{m+1}{2}\right)\right\}, & \text{se} \quad i = \frac{m+1}{2}. \end{cases} \tag{4.12}$$

No caso do último nível, o $L_{v_\ell} = 2N-1$ a vizinhança de v_ℓ é dada por:

$$Nb(2N-1,i) = \begin{cases} \{(2(N-1),j)\}, & \text{se} \quad i = 1, \\ \{(2(N-1),j-1),(2(N-1),j)\}, & \text{se} \quad i = 2, \ldots, N, \\ \{(2(N-1),N)\}, & \text{se} \quad i = N. \end{cases} \tag{4.13}$$

4.3.1 Relação um vértice, sua posição e nível

Considere a aplicação

$$\begin{array}{rcl} \phi : V(\mathcal{G}_n) & \longrightarrow & \mathbb{Z} \times \mathbb{Z} \\ v & \longmapsto & (L_v, p_v), \end{array}$$

em que L_v é o nível no qual o vértice v está localizado, isto é, a ordem da linha de isoestado relativa à S_0 e p_v é a posição do vértice v no nível L_v contada da esquerda para a direita na ordem de rotulação dos vértices de $\mathcal{G}_N$. Desta forma, ϕ relaciona o vértice v_ℓ, sua posição p_{v_ℓ} e nível L_k, ou seja, representa a sua imagem por $\phi(v_\ell) = (L_{v_\ell}, p_{v_\ell})$.

Para k par, tem-se:

$$\ell = p_{v_\ell} + \sum_{n=1}^{\frac{k}{2}-1} 2n + \frac{k}{2} \quad \therefore \quad 4\ell = 4p_{v_\ell} + k^2 \tag{4.14}$$

Para k ímpar e $\ell \geq 3$:

$$\ell = p_{v_\ell} + \sum_{n=1}^{\frac{k-1}{2}} 2n \quad \therefore \quad 4\ell + 1 = 4p_{v_\ell} + k^2. \tag{4.15}$$

Dado o nível L_k, as posições dos vértices respeitam as seguintes desigualdades:

$$1 \leq p \leq \left\lceil \frac{k}{2} \right\rceil,$$

em que $\lceil x \rceil$ é a função teto de x, ou seja, é o menor inteiro maior ou igual a x.

Logo,

$$4 \leq 4p_{v_\ell} \leq 4 \left\lceil \frac{k}{2} \right\rceil. \tag{4.16}$$

Fazendo a substituição de 4.16 em 4.14, obtém-se:

$$4 + k^2 \leq 4\ell \leq 4 \left\lceil \frac{k}{2} \right\rceil + k^2$$

agora, de posse da primeira desigualdade e fazendo as devidas simplificações, tem-se:

$$4 + k^2 \leq 4\ell \therefore k^2 \leq 4(\ell - 1) \therefore k \leq \sqrt{4\ell - 4}. \tag{4.17}$$

Dado um k inteiro e par e utilizando a segunda desigualdade de 4.17, tem-se:

$$4\ell \leq 4 \left\lceil \frac{k}{2} \right\rceil + k^2 = 2k + k^2 \quad \therefore \quad k^2 + 2k - 4\ell \geq 0 \ \therefore \ k \geq \sqrt{4\ell + 1} - 1. \tag{4.18}$$

Fazendo a substituição da primeira desigualdade de 4.16 em 4.15, obtêm-se:

$$4 + k^2 \leq 4\ell + 1 \leq 4\left\lceil \frac{k}{2} \right\rceil + k^2, \tag{4.19}$$

e, usando a primeira desigualdade acima, tem-se;

$$4 + k^2 \leq 4\ell + 1 \quad \therefore \quad k^2 \leq 4\ell - 3 \quad \therefore \quad k \leq \sqrt{4\ell - 3}. \tag{4.20}$$

Agora, usando a segunda desigualdade, obtém-se:

$$4\ell \leq 4\left\lceil \frac{k}{2} \right\rceil + k^2 = 4\left(\frac{k+1}{2}\right) + k^2 \ \ \therefore \ k \geq 2\sqrt{\ell} - 1 \tag{4.21}$$

Assim, para k qualquer (par ou ímpar) usando 4.17 e 4.20, tem-se:

$$k \leq \max\left\{\lfloor\sqrt{4\ell - 4}\rfloor, \lfloor\sqrt{4\ell - 3}\rfloor\right\}, \tag{4.22}$$

em que $\lfloor x \rfloor$ é a função teto de x,ou seja, é o maior inteiro, menor ou igual a x, e usando 4.18 e 4.21, tem-se:

$$k \geq \min\left\{\lceil\sqrt{4\ell + 1} - 1\rceil, \lceil 2\sqrt{\ell} - 1\rceil\right\} \tag{4.23}$$

Proposição 4.3.1. *Dado um $\ell \in \mathbb{Z}^+$, tem-se:*

$$\max\left\{\lfloor\sqrt{4\ell - 4}\rfloor, \lfloor\sqrt{4\ell - 3}\rfloor\right\} = \min\left\{\lceil\sqrt{4\ell + 1} - 1\rceil, \lceil 2\sqrt{\ell} - 1\rceil\right\}. \tag{4.24}$$

De fato, pela Proposição 4.3.1 e as Equações 4.22 e 4.23, obtém-se que k é o nível do vértice v_ℓ. Para obter a posição p_{v_ℓ}, utilizamos a Equação 4.14 e 4.15 e os valores de ℓ e k.

Exemplo 4.3.2. Considere o vértice v_{50}, ou seja $\ell = 50$. Temos que o nível de tal vértice é:

$$L_{50} = \max\{\lfloor\sqrt{4 \times 50 - 4}\rfloor, \lfloor\sqrt{4 \times 50 - 3}\rfloor\} = 14.$$

A posição de v_{50} no nível 14 é $p_{v_{50}} = \dfrac{4 \times 50 - 14^2}{4} = 1$. Assim, $\phi(v_{50}) = (14, 1)$. De (4.11) temos que $Nb(v_{50}) = \{(13, 1), (15, 1), (15, 2)\}$ e

$$\phi^{-1}(13, 1) = \frac{4 \times 1 + 13^2 - 1}{4} = 43 \quad \Rightarrow \quad v_{43} \text{ está no nível 13, primeira posição.}$$
$$\phi^{-1}(15, 1) = \frac{4 \times 1 + 15^2 - 1}{4} = 57 \quad \Rightarrow \quad v_{57} \text{ está no nível 15, primeira posição.}$$
$$\phi^{-1}(15, 2) = \frac{4 \times 2 + 15^2 - 1}{4} = 58 \quad \Rightarrow \quad v_{58} \text{ está no nível 15, segunda posição.}$$

Ou seja, $Nb(v_{50}) = \{v_{43}, v_{57}, v_{58}\}$ é a vizinhança do vértice v_{50}.

Nesta Seção, conseguiu-se demonstrar que os procedimentos de construção do grafo de estados mistos estão bem definidos, ou seja, dado um valor N que representa o número de partições do espaço de estados, tanto vértices, quanto arestas de $\mathcal{G}_N$ seguem uma rotulação bem definida. Na próxima Seção, irá se demonstrar o procedimento de se atribuir para cada vértice de $\mathcal{G}_N$ suas coordenadas (baricêntricas) relativas a S_0, S_1 e S_2.

4.4 Coordenadas dos vértices

Para cada subtriângulo $\mathcal{T}_i$, $i = 1, \ldots, N^2$ gerado, existe um vértice no baricentro de cada um desses substriângulos. As coordenadas baricêntricas de um vértice de $\mathcal{G}_\mathcal{N}$ é dada pela igual ponderação das coordenadas dos vértices do subtriângulo onde esta localizado. Já as coordenadas de cada vértice de um subtriângulo podem ser definidas pelas linhas de isoestados que formam cada um destes subtriângulos, a partir disso, tem-se as coordenadas desses vértices relativas aos estados puros.

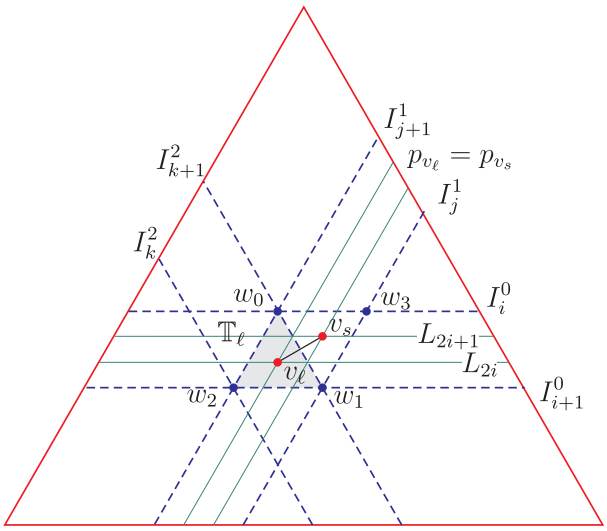


Figura 13 – Relação entre os vértices e as linhas de isoestados. Fonte: Autora.

Seja I^r_j a j-ésima linha de isoestado relativa a S_r, $r = 0, 1, 2$. Assim se N é o valor definido para o inverso do quantil inicial, tem-se

$$I^0_i = \left\{ \left(\frac{N-i}{N}, x_1, x_2 \right) \,:\, x_1 + x_2 = 1 - (N-i)/N \right\}, \quad i = 1, 2, \ldots, N, \tag{4.25}$$

$$I^1_j = \left\{ \left(x_0, \frac{N-j}{N}, x_2 \right) \,:\, x_0 + x_2 = 1 - (N-j)/N \right\}, \quad j = 1, 2, \ldots, N, \tag{4.26}$$

e

$$I^2_k = \left\{ \left(x_0, x_1, \frac{N-k}{N} \right) \,:\, x_0 + x_1 = 1 - (N-k)/N \right\}, \quad k = 1, 2, \ldots, N. \tag{4.27}$$

Seja v_ℓ um vértice de $\mathcal{G}_\mathcal{N}$, no nível $L_{v_\ell} = 2i + 1$ (ímpar!), $i = 0, 1, \ldots, 2N$, na posição $p_{v_\ell} = m$. O subtriângulo $\mathcal{T}_\ell$ é delimitado por I^0_i e I^0_{i+1}, ou seja, pelas linhas de isoestados:

$$I^0_i = \left(\frac{N-i}{N}, x_1, x_2 \right), \qquad e \qquad I^0_{i+1} = \left(\frac{N-i-1}{N}, x_1, x_2 \right)$$

Dado um vértice v_ℓ, num nível $L_{v_\ell} = 2i$ $i = 1, \ldots, 2N$ com $i = \frac{L v_\ell}{2}$, as mesmas conclusões devem ser feitas.

Dado o estado puro S_1, tem-se que para $\phi(v_\ell) = (2i + 1, p)$, ou $\phi(v_\ell) = (2i, p)$, existe a seguinte sequência de linhas e isoestados e posições nos níveis: $\ldots, I^1_{N-p+1}, \phi(v_\ell) = (2i + 1, p), \phi(v_s) = (2i, p), I^1_{N-p}, \ldots$, em que $i = 0, 1, \ldots, N$ e v_ℓ e v_s são vizinhos com $s = \phi^{-1}(2i, j)$. Logo o subtriângulo $\mathcal{T}_\ell$ é delimitado pelas isolinhas I^1_{N-p+1} e I^1_{N-p} e por

$$I^1_{N-p+1} = \left(x_0, \frac{N-(N-p+1)}{N}, x_2 \right) = \left(x_0, \frac{p-1}{N}, x_2 \right), \qquad e$$

$$I^1_{N-p} = \left(x_0, \frac{N-(N-p)}{N}, x_2\right) = \left(x_0, \frac{p}{N}, x_2\right)$$

De posse dessas definições, pode-se calcular as coordenadas dos vértices ω_1, ω_2 e ω_3 do subtriângulo $\mathcal{T}_\ell$. Como ω_1 pertence à linha de isoestado I^0_i, I^1_{j+1} e I^2_{k+1}, em que $i = \frac{L_{v_\ell}}{2}$ (ou $i = \frac{L_{v_\ell}-1}{2}$) e $j = N - p_{v_\ell}$, tal vértice tem coordenadas:

$$\omega_0 = \left(\frac{N-i}{N}, \frac{N-j-1}{N}, \frac{N-k-1}{N}\right).$$

No caso em que $L_{v_\ell} = 2i+1$, ou seja, $i = (L_{v_\ell}-1)/2$ e $j = N-p$, tem-se

$$\omega_0 = \left(1 - \frac{L_{v_\ell}-1}{2N}, \frac{p-1}{N}, S_2(w_0)\right),$$

em que

$$S_2(\omega_0) = 1 - \left(1 - \frac{L_{v_\ell}-1}{2N} + \frac{p-1}{N}\right) = \frac{L_{v_\ell} - 2p_{v_\ell} + 1}{2N},$$

Então,

$$\omega_0 = \left(1 - \frac{L_{v_\ell}-1}{2N}, \frac{p-1}{N}, \frac{L_{v_\ell} - 2p_{v_\ell}+1}{2N}\right) \tag{4.28}$$

para $\ell = 0\ldots, N-1$ e $j = 0, \ldots, N-1$ (em que v_ℓ está em um nível ímpar).

Dado que ω_1 pertence à linha de isoestado I^0_{i+1} e I^1_j e está sobre a mesma linha de isoestado I^2_k, logo ω_1 tem a terceira coordenada igual a de ω_0.

Então,

$$\omega_1 = \left(1 - \frac{L_{v_\ell}+1}{2N}, p_{v_\ell}, \frac{L_{v_\ell} - 2p_{v_\ell}+1}{2N}\right) \tag{4.29}$$

Agora, ω_2 está sobre a mesma linha de isoestado I^1_j que ω_1 e mesma linha de isoestado I^0_i de ω_2, e sobre I^2_k. Portanto,

$$\omega_2 = \left(1 - \frac{L_{v_\ell}+1}{2N}, \frac{p-1}{N}, \frac{L_{v_\ell} - 2p_{v_\ell}}{2N}\right). \tag{4.30}$$

A última coordenada de ω_2 foi obtida fazendo $1 - S_0(\omega_2) - S_1(\omega_2)$.

O vértice v_ℓ é uma ponderação de ω_1, ω_2 e ω_3:

$$v_\ell = \frac{1}{3}\omega_0 + \frac{1}{3}\omega_1 + \frac{1}{3}\omega_2 = \left(1 - \frac{3L_{v_\ell}+1}{6N}, \frac{3p_{v_\ell}-2}{3N}, \frac{3L_{v_\ell} - 6p_{v_\ell}+5}{6N}\right). \tag{4.31}$$

Para $L_{v_s} = 2i$, $i = 1, \ldots, N-1$, e $j = N - p_{v_s}$, tem-se

$$\omega_0 = \left(1 - \frac{L_{v_s}}{2N}, \frac{p_{v_s} - 1}{N}, \frac{L_{v_s} - 2p_{v_s} + 2}{2N}\right) \tag{4.32}$$

$$\omega_1 = \left(1 - \frac{L_{v_s} + 2}{2N}, \frac{p_{v_s}}{N}, \frac{L_{v_s} - 2p_{v_s} + 2}{2N}\right) \tag{4.33}$$

$$\omega_2 = \left(1 - \frac{L_{v_s}}{2N}, \frac{p_{v_s}}{N}, \frac{L_{v_s} - 2p_{v_s}}{2N}\right). \tag{4.34}$$

Das três equações acima (veja Algoritmo 3), tem-se

$$v_s = \frac{1}{3}w_0 + \frac{1}{3}w_1 + \frac{1}{3}w_2 = \left(1 - \frac{3L_{v_s} + 2}{6N}, \frac{3p_{v_s} - 1}{3N}, \frac{3L_{v_s} - 6p_{v_s} + 4}{6N}\right). \tag{4.35}$$

Nesta seção, foi realizada uma análise geométrica precisa dos vértices do grafo, o qual possui uma estrutura combinatória associada à sua geometria. Como resultado, foram obtidas as coordenadas de cada vértice do grafo, dando-lhe uma representação espacial e um sentido geométrico.

4.5 Fluxo discretizado sobre o Grafo de estados mistos

Para obter-se a função de probabilidade condicional que determina a dinâmica do espaço de estados $\mathcal{T}$ utiliza-se a Equação (4.2) que satisfaz (4.5) e suas hipóteses.

Para $n, k = 0, 1, 2, \ldots$ e $i, j = 1, \ldots, N^2$, denote por

$$p_{ij}^{nk} = p(n + k, v_j \,|\, n, v_i), \tag{4.36}$$

a probabilidade do sistema estar num estado representado por v_i no instante n e esse sistema atingir em k passos o estado representado pelo vértice v_j, tal que seja válida.

Afim de se definir um processo de difusão discreto, consideram-se as seguintes propriedades para p_{ij}^{nk}:

1. Propriedade markoviana:

$$p(m, v_j \,|\, n, v_i) = \sum_{v_k \in V(\mathcal{G}_N)} \sum_{\ell=n}^{m} p(\ell, v_k \,|\, n, v_i) \cdot p(m, v_j \,|\, \ell, v_k). \tag{4.37}$$

2. Definição do coeficiente de deriva[1], $b^r(m, v_i)$ e o coeficiente de difusão, $a^{rs}(m, v_i)$:

$$b^r(m, v_i) = \sum_{v_j \in Nb(v_i)} [S^r(v_i) - S^r(v_j)] \cdot p_{ij}^{m-1,1}, \ r = 0, 1, 2, \tag{4.38}$$

em que $Nb(v_i)$ é conjunto de vizinhos de v_i, ou seja, o conjunto de vértices adjacentes a v_i. Também, para $r, s = 0, 1, 2$

$$2a^{rs}(m, v_i) = \sum_{v_j \in Nb(v_i)} [S^r(v_i) - S^r(v_j)][S^s(v_i) - S^s(v_j)] \cdot p_{ij}^{m-1,1}. \tag{4.39}$$

O vetor $(b^r(m, v_l))_{r=0,1,2}$ relaciona-se com a média dos estados vizinhos de v_l dados pelas probabilidades de transição em um passo, no instante m. Esta média é dada por (4.7). De forma análoga, a matriz $\left(a^{rs}(m, v_l)\right)_{rs}$ relaciona-se à covariância entre as coordenadas dos estados puros.

A equação de Kolmogorov para o caso dos grafos de estados mistos podem ser escrita como

$$a^{rs}(m-1, v_\ell)\frac{\Delta^2 p_{\ell i}^{m-1,1}}{\Delta_{\ell i}S^r \Delta_{\ell i}S^s} + b^r(m-1, v_\ell)\frac{\Delta p_{\ell i}^{m-1,1}}{\Delta_{\ell i}S^r} = -\Delta p_{\ell i}^{m-1,1}, \quad v_\ell \in V(\mathcal{G}_N), \tag{4.40}$$

em que $\Delta_{\ell i}S^r = S^r(v_\ell) - S^r(v_i)$, $\Delta p_{\ell i}^{m-1,1} = p_{\ell i}^{m-1,1} - p_{i\ell}^{m,-1}$ e $\Delta p_{\ell i}^{m-1,1} = -\Delta p_{i\ell}^{m,-1}$.

Utilizando a expressão acima e a reescrevendo, tem-se:

$$\frac{\Delta^2 p_{\ell i}^{m-1,1}}{\dfrac{\Delta_{\ell i}S^r \Delta_{\ell i}S^s}{a^{rs}(m-1, v_\ell)}} + \frac{\Delta p_{\ell i}^{m-1,1}}{\dfrac{\Delta_{\ell i}S^r}{b^r(m, v_\ell)}} = -\Delta p_{\ell i}^{m-1,1}, \tag{4.41}$$

Logo, para todo vértice $v_\ell \in V(\mathcal{G}_N)$, tem-se:

$$\sum_{r,s=0}^{2}\left[\sum_{v_i \in Nb(v_\ell)} \frac{(p_{\ell i}^{m-1,1} - p_{i\ell}^{m,-1})^2}{\dfrac{[S_r(v_\ell) - S_r(v_i)][S_s(v_\ell) - S_s(v_i)]}{\sum_{v_j \in Nb(v_\ell)}[S_r(v_\ell) - S_r(v_j)][S_s(v_\ell) - S_s(v_j)]}} \cdot p_{\ell i}^{m-1,1}\right]$$
$$+\sum_{r=0}^{2}\left[\sum_{v_i \in Nb(v_\ell)} \frac{(p_{\ell i}^{m-1,1} - p_{i\ell}^{m,-1})}{\dfrac{[S_r(v_\ell) - S_r(v_i)]}{\sum_{v_j \in Nb(v_\ell)}[S_r(v_\ell) - S_r(v_j)]}} \cdot p_{\ell i}^{m-1,1}\right] = \sum_{v_i \in Nb(v_\ell)} p_{i\ell}^{m,-1} \tag{4.42}$$

[1] Mais comumente, denomina-se coeficiente *drift*.

A Equação 4.42 acima relacionada à Equação 4.9 do Teorema de Kolmogorov, envolve as probabilidades de mudança de um vértice para algum de seus vizinhos numa dinâmica sobre o grafo $\mathcal{G}_N$ e será justamente utilizada para se determinar estas probabilidades em simulações apresentadas no Capítulo 5 .

4.6 Algoritmo

Cormen et al. (2022) definem de maneira generalizada um *algoritmo* como sendo qualquer procedimento computacional que toma algum valor ou conjunto de valores, como entrada e produz algum valor ou conjunto de valores, como saída. Um algoritmo é, portanto, uma sequência de passos computacionais que transformam a entrada em saída.

Também pode-se visualizar um algoritmo como uma ferramenta para resolver problemas computacionais bem especificado. O enunciado do problema especifica em termos gerais o relacionamento de entrada/saída desejada e o algoritmo descreve um procedimento computacional específico para alcançar essa relação entrada/saída.

Rudimentalmente falando, um algoritmo é um conjunto de passos necessários para realizar uma tarefa. Sua implementação pode ser feita por um computador, por outro tipo de autômato ou mesmo por um ser humano. A maneira mais simples de se pensar um algoritmo é por uma lista de procedimentos bem definida, na qual as instruções são executadas passo a passo a partir do começo da lista, uma ideia que pode ser facilmente visualizada através de um fluxograma. Cada campo da ciência possui seus próprios problemas e respectivos algoritmos adequados para resolvê-los. Exemplos clássicos são algoritmos de busca, de ordenação, de análise numérica, de teoria de grafos, de manipulação de cadeias de texto, de geometria computacional, de análise combinatória, de aprendizagem de máquina, de criptografia, de compressão de dados e de interpretação de texto.

Para ser possível construir um algoritmo que resolva a dinâmica de um grafo, serão necessário algumas etapas que abrangem:

- Para descrever um sistema físico que possui três estados puros, denotados por S_0, S_1 e S_2, ou combinações convexas de estados. Também será necessário, definir o espaço de estados, fazendo a descrição do conjunto de equação que o descreveria;
- Para gerar um sistema de coordenadas que localiza o estado atual do sistema relativamente aos três estados puros, será especificado uma *n-upla* de escalares para

cada ponto do espaço do espaço $(n-1)$-*dimensional*. O sistema de coordenadas utilizados será o sistema de coordenadas baricêntricas de um triângulo equilátero cujo os vértices são os estados puros, denotados por S_0, S_1 e S_2.

4.6.1 Algoritmo de definição do grafo de estados mistos

Dado um espaço de estados $\mathcal{T}$, o usuário escolhe, de forma arbitrária, o número de quantis, denominado de N onde $N \in \mathbb{N}$. Deste modo é definido o *quantil* de discretização, $p = N^{-1}$ e posteriormente é definido o grafo de estados mistos $\mathcal{G}_{\mathcal{N}}$ seguindo o Algoritmo 1 abaixo. O grafo proveniente do resultado deste Algoritmo é ilustrado pela Figura 14.

Algoritmo 1: Procedimento de construção do grafo $\mathcal{G}_N$.

▷ Leia o número de quantis N;
▷ Divida cada lado do triângulo $\mathcal{T}$ em N partes iguais (cada ponto representa um p-quantil das linhas $(0, x_1, x_2)$, $(x_0, 0, x_2)$ e $(x_0, x_1, 0)$);
▷ Use cada p-quantil para determinar linhas de *isoestados* relativamente a cada estado puro. Estas linhas são paralelas aos lados dos triângulos e chamaremos de isolinhas;
▷ Após isso serão formados N^2 triângulos $\mathcal{T}_i$. Em cada baricentro de tais $\mathcal{T}_i$'s considere um vértice e una dois vértices por uma aresta se, e somente se, dois dos dois triângulos correspondentes tiverem um lado em comum.

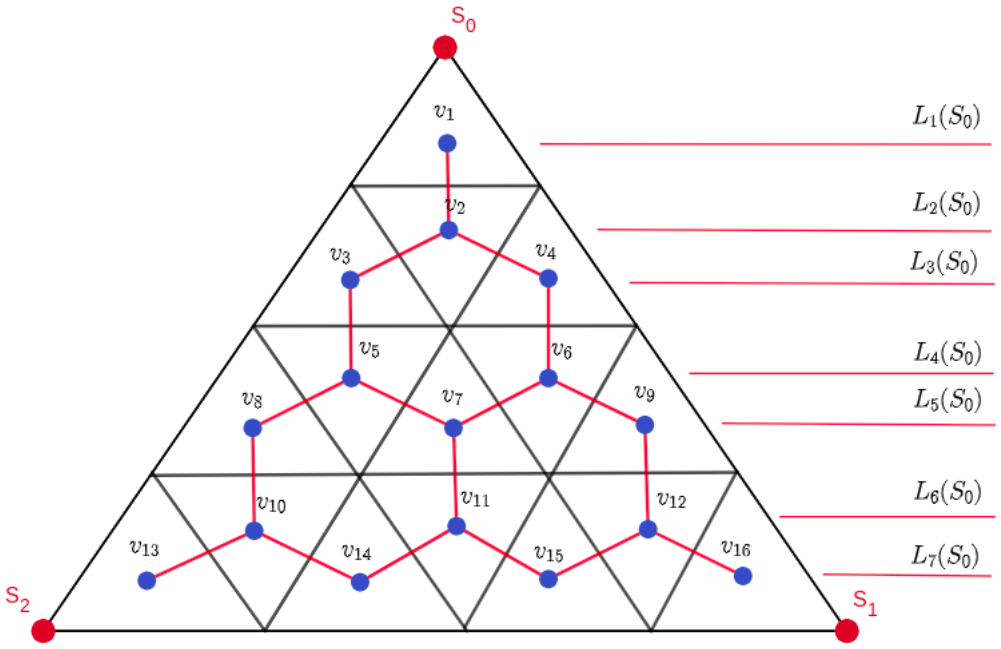


Figura 14 – Representação do grafo de estados mistos $\mathcal{G}_4$ do sistema S. Fonte: Autora.

A discretização é feita considerando apenas as coordenadas baricêntricas dos vértices dos triângulos, ou seja, $(v_1, v_2, v_3, ...v_{16})$ as quais representa o estado atual do sistema, o v_ℓ com $\ell = (1, 2, ...N^2) \in \mathbb{N}$. A dinâmica é feita considerando a transição do estado atual, v_1, no instante σ para o instante $\sigma + 1$. O estado pode permanecer em v_1 ou mudar para v_2, para essa transição denominamos de transição suave nesse espaço discreto e as coordenadas do vetor de estados v_ℓ é denotada por $v_\ell = (S_0(v_\ell), S_1(v_\ell), S_2(v_\ell))$. A quantidade de níveis é dada pela Proposição 4.2.1, para $N = 4$, tem-se sete níveis representados respectivamente por $L_1(S_0), L_2(S_0), \cdots, L_7(S_0)$. Utilizando a Definição 2.5.7 para $N = 4$ pode-se gerar a matriz de adjacência desse grafo.

4.6.2 Algoritmo da construção da matriz de adjacência do grafo de estados mistos

O Algoritmo 2 a seguir é utilizado para construir a matriz de adjacência do grafo $\mathcal{G}_N$, atribuindo a vizinhança de cada vértice, como por exemplo, aquele representado pela Figura 14.

4.6.3 Algoritmo das coordenadas dos vértices do grafo de estados mistos

O Algoritmo 3 abaixo será utilizado para atribuir as coordenadas de cada vértice relativas aos estados puros (S_0, S_1, S_2). Relembrando que cada vértice foi colocado dentro do baricentro de um dos subtriângulos $\mathcal{T}_i, i = 1, \cdots N^2$ representados pela Figura 14. O vértice v_ℓ pode ser representado pelas coordenadas $v_\ell = (S_0, S_1, S_2)$.

4.6.4 Algoritmo da dinâmica

O Algoritmo 4 gera a caminhada no grafo durante o tempo estabelecido. Gera uma dinâmica entre os vértices utilizando os resultados dos Algoritmos 2 e 3 e com probabilidades geradas de forma aleatória ou arbitrária.

Algoritmo 2: Matriz de adjacência do grafo $\mathcal{G}_{\mathcal{N}}$. Fonte: Autora

Entrada: N = valor do quantil dado.
Saída: A matriz de adjacência do grafo
início
  $L_{v_1} = 1$; $p_{v_1} = 1$; $Matriz(1,2) = 1$;
  repita
    $L_{v_\ell} = \max\{\lfloor\sqrt{4\ell-4}\rfloor, \lfloor\sqrt{4\ell-3}\rfloor\}$ `% Calcula o nível do vértice` v_ℓ;
    se L_{v_ℓ} *é par* **então**
      $p_{v_\ell} = [4\ell - (L_{v_\ell})^2]/4$;
      $i = [4p_{v_\ell} + (L_{v_\ell} - 1)^2 - 1]/4$;
      $j = [4p_{v_\ell} + (L_{v_\ell} + 1)^2 - 1]/4$;
      $k = [4p_{v_\ell} + (L_{v_\ell} + 1)^2 + 3]/4$;
      } `% Vizinhos de` v_ℓ;
      $Matriz(\ell, i) = 1$; $Matriz(\ell, j) = 1$; $Matriz(\ell, k) = 1$;
    senão
      $p_{v_\ell} = [4\ell - (L_{v_\ell})^2 + 1]/4$;
      se $L_{v_\ell} \neq 2N - 1$ **então**
        se $p_{v_\ell} = 1$ **então**
          $i = [4 + (L_{v_\ell} - 1)^2]/4$; $j = [4 + (L_{v_\ell} + 1)^2]/4$;
          $Matriz(\ell, i) = 1$; $Matriz(\ell, j) = 1$;
        senão
          se $p_{v_\ell} = (L_{v_\ell} + 1)/2$ **então**
            $i = [2(L_{v_\ell} - 1) + (L_{v_\ell} - 1)^2]/4$; $j = [2(L_{v_\ell} + 1) + (L_{v_\ell} + 1)^2]/4$;
            $Matriz(\ell, i) = 1$; $Matriz(\ell, j) = 1$;
          senão
            $i = [4(p_{v_\ell} - 1) + (L_{v_\ell} - 1)^2]/4$; $j = [4p_{v_\ell} + (L_{v_\ell} - 1)^2]/4$;
            $k = [4p_{v_\ell} + (L_{v_\ell} + 1)^2]/4$;
            $Matriz(\ell, i) = 1$; $Matriz(\ell, j) = 1$; $Matriz(\ell, k) = 1$;
          fim
        fim
      senão
        se $p_{v_\ell} = 1$ **então**
          $i = [4 + (L_{v_\ell} - 1)^2]/4$; $Matriz(\ell, i) = 1$;
        senão
          se $p_{v_\ell} = (L_{v_\ell} + 1)/2$ **então**
            $i = [2(L_{v_\ell} - 1) + (L_{v_\ell} - 1)^2]/4$; $Matriz(\ell, i) = 1$;
          senão
            $i = [4(p_{v_\ell} - 1) + (L_{v_\ell} - 1)^2]/4$; $j = [4p_{v_\ell} + (L_{v_\ell} - 1)^2]/4$;
            $Matriz(\ell, i) = 1$; $Matriz(\ell, j) = 1$;
          fim
        fim
      fim
    fim
  até $\ell = N^2$;
fim

Algoritmo 3: Coordenadas dos vértices do grafo $\mathcal{G}_N$ relativas aos estados puros S_0, S_1 e S_2. Fonte: Autora.

Entrada: N = valor do quantil ;
L_{v_ℓ}, p_{v_ℓ}, nível e posição do vértice $v_\ell \in V(\mathcal{G}_\mathcal{N})$
Saída: As coordenadas baricêntricas de cada vértice v_ℓ de $\mathcal{G}_\mathcal{N}$
início
  repita
    se L_{v_ℓ} *é ímpar* **então**
      $v_\ell = \left(1 - \frac{3L_{v_\ell}+1}{6N}, \frac{3p_{v_\ell}-2}{3N}, \frac{3L_{v_\ell}-6p_{v_\ell}+5}{6N}\right)$;
    senão
      $v_\ell = \left(1 - \frac{3L_{v_\ell}+2}{6N}, \frac{3p_{v_\ell}-1}{3N}, \frac{3L_{v_\ell}-6p_{v_\ell}+4}{6N}\right)$;
    fim
  até $\ell = N^2$;
fim

Algoritmo 4: Algoritmo de simulação de trajetórias no grado de estados. Fonte: Autora.

início
  repita
    $\ell = 1$ **repita**
      $u = random(0, 1)$;
      se $0 < u <= p_{\ell\ell}$ **então**
        $trajetoria[dia, m] = dados[\ell, 1]$
      senão
        se $p_{\ell\ell} < u \leq (p_{\ell\ell} + p_{\ell i}$ **então**
          $trajetoria[dia, m] = dados[\ell, 1]$
      fim
      senão
        se $(p_{\ell\ell} + p_{\ell i} < u \leq p_{\ell\ell} + p_{\ell i} + p_{\ell,j})$ **então**
          $trajetoria[dia, m] = dados[\ell, 2]$
      fim
      senão
        se $(p_{\ell\ell} + p_{\ell i} + p\ell, j < u \leq 1$ **então**
          $trajetoria[dia, m] = dados[\ell, 3]$;
      fim
      $\ell = trajetoria[dia, m]$
    até $dia = tempo$;
  até $m = 1000$;
fim

Para o Algoritmo 4, as variáveis são definidas da seguinte forma:

1. **u** é uma variável aleatória uniforme entre 0 e 1, sorteada a cada iteração do *loop*.
2. $\mathbf{p_{\ell\ell}}$ é a probabilidade de se permanecer no vértice v_ℓ.
3. $\mathbf{p_{\ell s}}$ é a probabilidade de se mudar do vértice v_ℓ para o vértice v_s, $s = i, j$, ou k.
4. **dados**$[\ell, \mathbf{coluna}]$ é um vetor bidimensional que recebe como entrada o vértice atual e uma coluna fixa. Esse valor de coluna é referente ao local no qual é armazenado um dos vizinhos do vértice ℓ.
5. **trajetoria**$[\mathbf{dia}, \mathbf{m}]$ é um vetor bidimensional que recebe como entrada o valor de m (o número da simulação), o dia indicado e no qual está ocorrendo a dinâmica. Lembrando que esses valores são dados pois dois *loop* distintos e aninhados.

A simulação da dinâmica em estudo se dá da seguinte forma:

1. Inicia-se um *loop* com a quantidade de simulações a ser gerada;
2. Inicia-se um *loop* com a quantidade de dias ao qual se quer verificar a trajetória;
3. Para cada iteração do *loop*, é gerado de forma aleatória um número u entre 0 e 1;
4. É analisada as condições estabelecidas: Se u estiver no intervalo semi-aberto $]0, pll]$ o vértice permanece no mesmo local, ou seja, vértice atual = v_ℓ.
5. Se u estiver no intervalo semi-aberto $]p_{\ell\ell}, (p_{\ell\ell} + p_{\ell i})]$ o vértice atual muda para o seu vizinho **i**, ou seja, vértice atual é igual a v_i.
6. Se u estiver no intervalo semi-aberto $](p_{\ell\ell} + p_{\ell i}), (p_{\ell\ell} + p_{\ell i} + p_{\ell j})]$ o vértice atual muda para o seu vizinho v_j, ou seja, vértice atual é v_j.
7. Se u estiver no intervalo semi-aberto $](p_{\ell\ell} + p_{\ell i} + p_{\ell j}), 1]$ o vértice atual muda para o seu vizinho v_k, ou seja, vértice atual é o v_k.
8. Após o fim de cada iteração do *loop*, o valor de **l** é atualizado pelo valor dado por $trajetoria[dia, m]$.

Cada trajetória é calculada m vezes. Para as simulações neste trabalho usou-se para o Algoritmo 4, $m = 1000$. Os algoritmos dessa Seção serão implementos na linguagem *Python* , como apresentação nas próximas seções.

4.7 Código em *Python*

Nesta Seção serão apresentados alguns detalhes sobre os códigos de programação que foram usados para a efetivação dos algoritmos apresentados e nas simulações executadas.

4.7.1 Código em *Python* da matriz de adjacência do grafo de estados mistos

O código referente ao Anexo A é a implementação do Algoritmo 2, o qual é responsável pela construção da matriz de adjacência do grafo $\mathcal{G}_N$, cálculo da vizinhança do vértice v_ℓ, seus respectivos níveis L_{v_ℓ} e posições p_{v_ℓ}.

Este código foi totalmente construído na Linguagem *Python* utilizando as bibliotecas Numpy, Pandas, CSV e math

- A variável **N** corresponde ao quantil N definido pela Proposição 4.2.1;
- A variável **nivel_Lvl** corresponde ao nível L_{v_ℓ} definido pela Proposição 4.2.1;
- A variável **pos_vl** corresponde a posição, p_{v_ℓ}, do vértice v_ℓ definida pela Equação 4.16;
- A variável **matriz** corresponde a matriz de adjacência do grafo $\mathcal{G}_N$, $A = (a_{ij})_{nxn}$, e é definida pela Equação 2.5.7;
- A variável **l** corresponde ao vértice v_l;
- A variável **i**, caso exista, corresponde ao vértice vizinho v_i do vértice v_l e é dado pela Definição 2.5.6;
- A variável **j**, caso exista, corresponde ao vértice vizinho v_j do vértice v_l e é dado pela Definição 2.5.6;
- A variável **k**, caso exista, corresponde ao vértice vizinho v_k do vértice v_l e é dado pela Definição 2.5.6;
- A variável **vertices_vizinhos** corresponde a um arquivo .csv onde é armazenado os vértices v_ℓ e sua vizinhança para cada valor N de quantil definido.

4.7.2 Código em *Python* das coordenadas dos vértices do grafo de estados mistos

O código referente ao Anexo B é a implementação do Algoritmo 3, o qual é responsável pela obtenção das coordenadas baricêntricas do grafo $\mathcal{G}_N$, representado pela Figura 11.

Este código foi totalmente construído na Linguagem *Python* . Considere como válidas as definições dada por 4.7.1.

Tome $v_\ell = (S_0, S_1, S_2)$ representado pela 11,

- A variável **si0** corresponde a primeira coordenada baricêntrica do vértice v_ℓ, o S_0.
- A variável **si1** corresponde a segunda coordenada baricêntrica do vértice v_ℓ, o S_1.
- A variável **si2** corresponde a terceira coordenada baricêntrica do vértice v_ℓ, o S_2.

4.7.3 Código em *Python* da dinâmica da simulação do grafo de estados mistos

O código referente ao Anexo C descreve a dinâmica estocástica dos estados do sistema e é a implementação do Algoritmo 2, Algoritmo 3 em conjunto com a dinâmica estocástica dos estados e construção do desenho do grafo $\mathcal{G}_N$.

Este foi totalmente construído na Linguagem Python utilizando as bibliotecas Numpy, Pandas, CSV e math. Considere como válidas as definições dada por 4.7.1. No código referente ao Anexo C definimos 3 novas variáveis, **o,r,s** responsáveis pela dinâmica com o banco de dados real.

Tome $v_\ell = (S_0, S_1, S_2)$ representado pela 11,

- A variável **o** corresponde a função teto da primeira coordenada baricêntrica do vértice v_ℓ, o S_0, multiplicada pelo quantil **N**, ou seja, $o = \lceil N_0 S_0 \rceil$.
- A variável **r** corresponde a função teto da segunda coordenada baricêntrica do vértice v_ℓ, o S_1, multiplicada pelo quantil **N**, ou seja, $r = \lceil N_0 S_1 \rceil$.
- A variável **s** corresponde a função teto da terceira coordenada baricêntrica do vértice v_ℓ, o S_2, multiplicada pelo quantil **N**, ou seja, $s = \lceil N_0 S_2 \rceil$.

4.8 Banco de dados

4.8.1 Sistemas Gerenciadores de Banco de dados

Os Gerenciadores de Banco de dados, popularmente conhecidos como *SGBD* são softwares especializados para gerenciamento de bases de dados.

As principais funções de um SGBD são:

- Alteração da estrutura de campos;
- Cópia e eliminação de ficheiros;
- Inserção, remoção e criação de relações entre tabelas;
- Importação e exportação de dados entre bases de dados;
- Criação de chaves externas e primárias;
- Consultas nas tabelas;
- Criação de usuários com permissões de acesso.

O SGBD utilizado será o PostgreSQL em conjunto com a plataforma pgAdmin4.

4.8.2 Linguagem de Consulta Estruturada

A linguagem de consulta estruturada, popularmente conhecida como SQL (*Structured Query Language*), é a linguagem é uma das linguagens mais importantes para manipulação de registros em banco de dados relacionais.

A linguagem SQL possui uma organização estrutural baseadas em subdivisões dos seus comandos, cada qual divididos em 5 subconjuntos diferentes.

DML - Data Manipulation Language: Também conhecida como linguagem de manipulação de dados, esse subconjunto do SQL define os comandos usados para manipular os dados armazenados em um banco. Esse é um dos conjuntos mais utilizados, pois ele fornece operadores que nos permitem inserir, excluir e alterar os registros de uma tabela, por exemplo. Os comandos mais importantes desse subconjunto são: `INSERT`, `DELETE` e `UPDATE`.

DQL - Data Query Language: Também conhecida como linguagem de consulta de dados, esse subconjunto do SQL define o `SELECT`, comando essencial para consulta de dados armazenados no banco.

DDL - Data Definition Language: Também conhecida como linguagem de definição de dados, esse subconjunto do SQL define os comandos usados para gerenciar as estruturas do banco de dados. É o responsável por criar, atualizar e remover objetos da base, como tabelas e índices. Os comandos definidos pelo DDL são: `CREATE`, `DROP` e `ALTER`.

DCL - Data Control Language: Também conhecida como linguagem de controle de dados, esse subconjunto do SQL define os comandos para controle do acesso aos dados da base de dados. É o responsável por estabelecer restrições e permissões para o acesso ao banco. Os comandos definidos pelo DCL são: `GRANT` e `REVOKE`.

DTL - Data Transaction Language ou TCL (Transaction Control Language): Também conhecida como linguagem de transação de dados, esse subconjunto do SQL define os comandos que utilizamos quando é necessário gerenciar transações feitas no banco. É o responsável por iniciar, confirmar e desfazer alterações. Os comandos definidos pelo DTL|TCL são: `COMMIT`, `BEGIN` e `ROLLBACK`.

Para esse caso, apenas serão utilizado os subconjuntos DQL e DDL.

4.8.3 PostgreSQL e pgAdmin4

PostgreSQL é um sistema gerenciador de banco de dados objeto-relacional, *open source*, baseado no POSTGRES e desenvolvido no Departamento de Ciências da Computação em Berkeley, Universidade da Califórnia. No PostgreSQL tudo que é criado é tratada como um objeto, por exemplo banco de dados, tabelas, *triggers*, *views*, etc. Já o pgAdmin4 é um software gráfico para administração do SGBD PostgreSQL.

4.8.4 Referência de Banco de dados

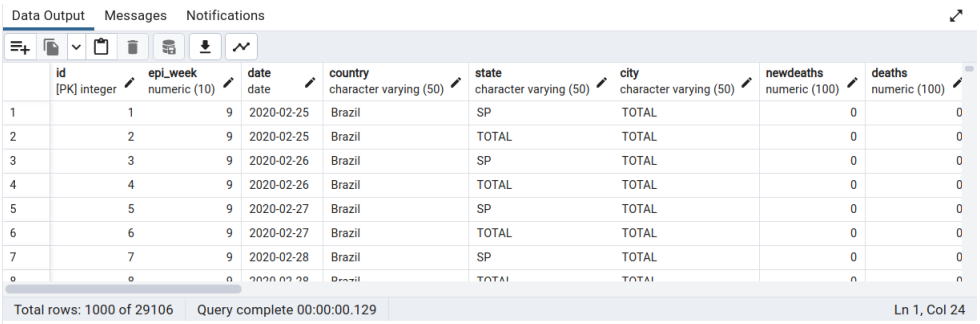
O banco de dados utilizado para análise foi o criado pelo Wesley Cota e se encontra disponível em Banco de dados COVID BR. O arquivo foi disponibilizado na extensão .CSV e importado para o pgAdmin4 pelo *script* abaixo.

4.8.4.1 DDL: Importando os dados para o pgAdmin4

Através do comando `COPY` é possível importar dados de um arquivo .CSV para uma tabela, criada pelo comando `CREATE TABLE`. Verifique o Anexo D.

Dada as devidas modificações, o mesmo processo se repete para inserção dos dados relacionados ao Algoritmo C. As tabelas provenientes desse Algoritmo foram denominadas de *verticesn10_x* sendo *10_x* o valor do quantil escolhido pelo usuário.

A saída do *script* do Anexo D deverá ser parecida com a Figura 15:


Data Output Messages Notifications

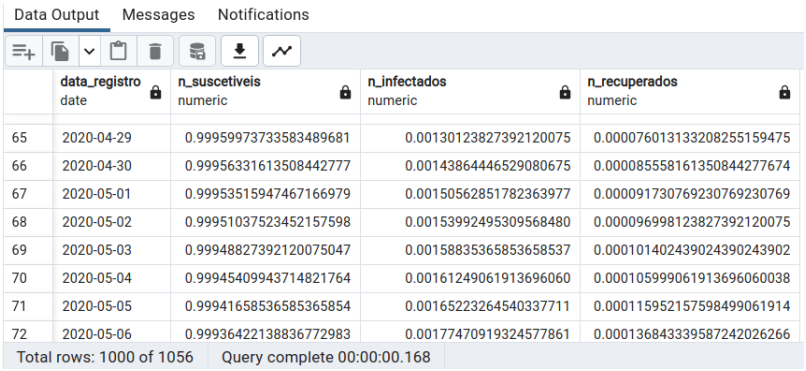
	id [PK] integer	epi_week numeric (10)	date date	country character varying (50)	state character varying (50)	city character varying (50)	newdeaths numeric (100)	deaths numeric (100)
1	1	9	2020-02-25	Brazil	SP	TOTAL	0	0
2	2	9	2020-02-25	Brazil	TOTAL	TOTAL	0	0
3	3	9	2020-02-26	Brazil	SP	TOTAL	0	0
4	4	9	2020-02-26	Brazil	TOTAL	TOTAL	0	0
5	5	9	2020-02-27	Brazil	SP	TOTAL	0	0
6	6	9	2020-02-27	Brazil	TOTAL	TOTAL	0	0
7	7	9	2020-02-28	Brazil	SP	TOTAL	0	0

Total rows: 1000 of 29106 Query complete 00:00:00.129 Ln 1, Col 24

Figura 15 – Saída do *script* D. Fonte: Autora.

4.8.4.2 DQL: Consultas no banco

A *query* abaixo é responsável por extrair as porcentagens de suscetíveis, infectados e recuperados da população total do Brasil, baseado nos dados disponíveis em Banco de dados COVID BR sendo a saída da *query* uma tabela com as respectivas porcentagens dia a dia. Um exemplo de saída do *script* do Anexo E é ilustrado pela Figura 16:


Data Output Messages Notifications

	data_registro date	n_suscetiveis numeric	n_infectados numeric	n_recuperados numeric
65	2020-04-29	0.99959973733583489681	0.00130123827392120075	0.00007601313320825515947 5
66	2020-04-30	0.99956331613508442777	0.00143864446529080675	0.00008555816135084427767 4
67	2020-05-01	0.99953515947467166979	0.00150562851782363977	0.00009173076923076923076 9
68	2020-05-02	0.99951037523452157598	0.00153992495309568480	0.00009699812382739212007 5
69	2020-05-03	0.99948827392120075047	0.00158835365853658537	0.00010140243902439024390 2
70	2020-05-04	0.99945409943714821764	0.00161249061913696060	0.00010599906191369606003 8
71	2020-05-05	0.99941658536585365854	0.00165223264540337711	0.00011595215759849906191 4
72	2020-05-06	0.99936422138836772983	0.00177470919324577861	0.00013684333958724202626 6

Total rows: 1000 of 1056 Query complete 00:00:00.168

Figura 16 – Saída do *script* C. Fonte: Autora.

5 Resultados

5.1 Aplicação do modelo com probabilidades arbitrárias

Nesta Seção será abordada a implementação do modelo de discretização para cenários fictícios com probabilidades arbitrárias geradas para cada vértice do grafo. De início, será utilizado o modelo com probabilidades arbitrárias, escolhidas de forma lógica em aderência à uma sistema ternário que considera um espaço de estados com 3 estados puros S_0, S_1 e S_2 respectivamente: suscetíveis, infectados e removidos ou recuperados.

Para a primeira análise e como forma de exemplificar de facilitar o entendimento do leitor, considera-se um sistema S, com 3 estados puros S_0, S_1 e S_2 e com valor de $N = 4$. O grafo referente a esse sistema pode ser visualizado na Figura 14 e sua construção se dá pelos Algoritmos 2, 3 acima descritos.

De posse do grafo e de todas as informações que o carregam, foram feitas simulações de possíveis trajetórias (caminhada do vértice). Note que para o caso específico tratado, a dinâmica da epidemia de uma doença, o COVID-19, o vértice do primeiro dia necessariamente será o vértice v_1, ou seja, $v_\ell = v_1$

Para esse caso, foram geradas $m = 1000$ simulações de possíveis dinâmicas para o grafo $\mathcal{G}_N$, com $N = 4$.

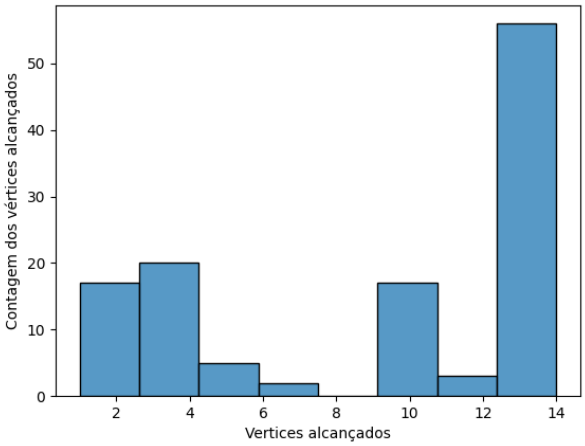


Fig. 17 – Histograma contendo a simulação da dinâmica para 730 dias com $N = 4$ e $m = 2$. Fonte: esta pesquisa.

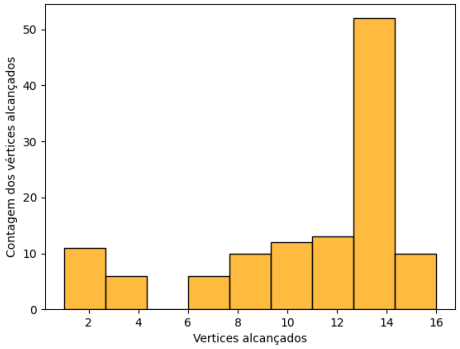


Fig. 18 – Histograma contendo a simulação da dinâmica para 730 dias com $N = 4$ e $m = 180$. Fonte: esta pesquisa.

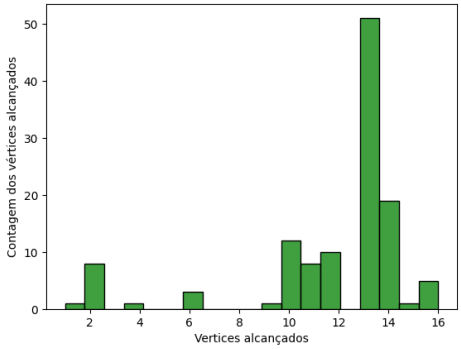


Fig. 19 – Histograma contendo a simulação da dinâmica para 730 dias com $N = 4$ e $m = 295$. Fonte: esta pesquisa.

Na Figura 17 foi plotado um histograma da simulação da trajetória com $m = 2$ onde o eixo y se refere a quantidade de vértices alcançados durante a trajetória e o eixo x os respectivos vértices. Analogamente, a Figura 18 e a Figura 19 são os histogramas para $m = 180$ e $m = 295$, respectivamente.

Nota-se até de forma visual, que para o caso com 16 vértices, o último vértice v_16 é atingido majoritariamente em ambas as simulações.

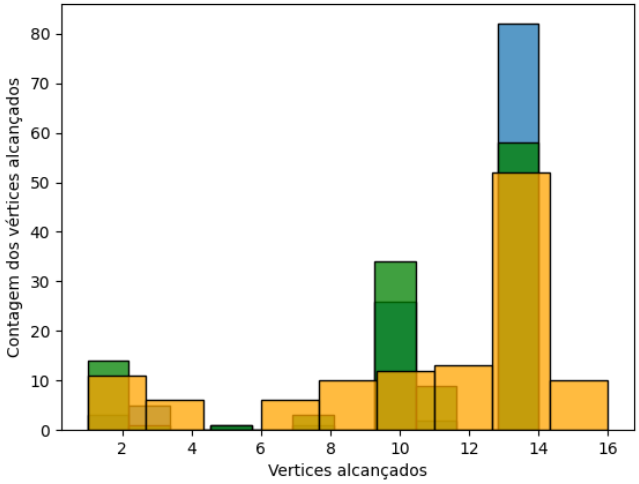


Figura 20 – Histogramas sobrepostos referente as simulações das trajetórias de 730 dias para um grafo com $N = 4$. Fonte: esta pesquisa.

A Figura 20 demonstra de forma clara a variabilidade de trajetórias para cada instância da simulação.

Ainda para $N = 4$, utiliza-se o gráfico box plot para demonstrar graficamente a variabilidade dos dados relativos a dinâmica. Para gerar esse gráfico, é feita uma relação entre os vértices alcançados na trajetória da dinâmica para os 730 dias e suas respectivas coordenadas baricêntricas. Vale salientar que nesse caso, as coordenadas baricêntricas do grafo representam respectivamente a porcentagem pela população total de suscetíveis, infectados e recuperados. Pela Figura 21 se verifica que a dinâmica percorre todos os vértices do grafo em apenas um mês.

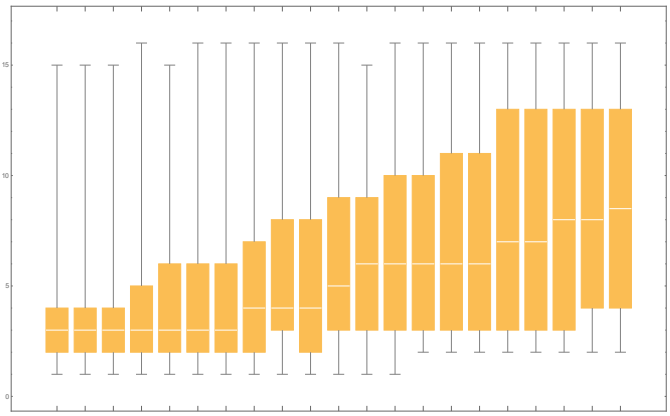


Figura 21 – Gŕafico boxplot para $N = 4$ dos dias 10 ao 30. Fonte: esta pesquisa. Fonte: esta pesquisa.

No Gráfico da Figura 22 é possível visualizar como se comporta as curvas relativas

as porcentagens de suscetíveis, infectados e recuperados durante as trajetórias simuladas. O que, de fato, se assemelha bastante ao esperado numa epidemia. Os números de suscetíveis diminui durante o decorrer do tempo, o de infectados aumenta e o de recuperados varia de forma comitente aos outros dois.

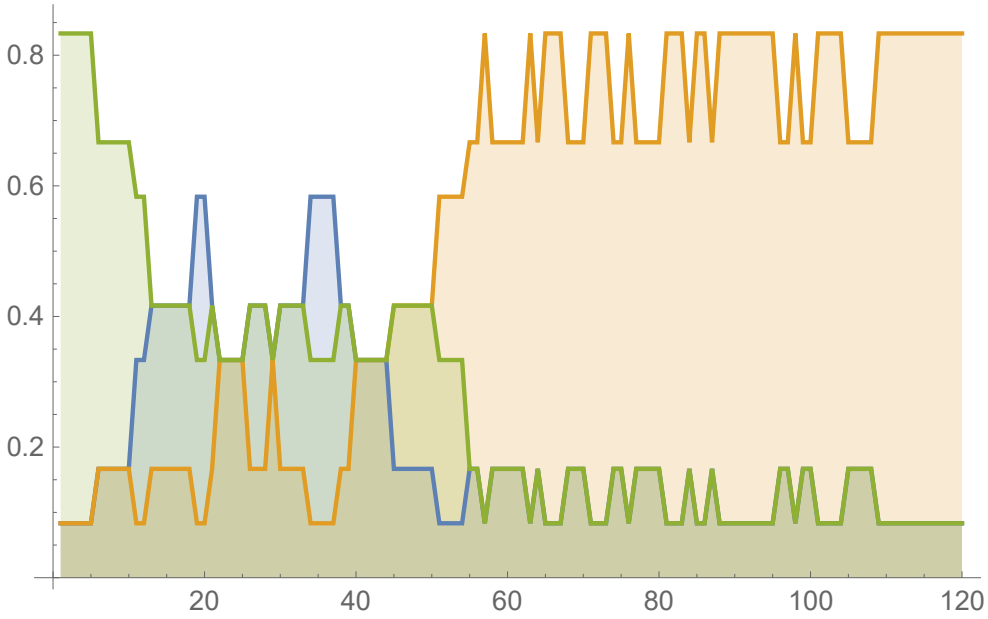


Figura 22 – Curvas das dinâmicas dos suscetíveis, infectados e recuperados para um grafo com $N = 4$. Verde: suscetíveis; Azul: infectados; Laranja: recuperados. Fonte: esta pesquisa.

Na Figura 22 a curva verde representa a porcentagem dos suscetíveis, a azul a porcentagem dos recuperados e a laranja a porcentagem dos infectados.

Um outro exemplo de aderência do modelo aos dados reais, utiliza-se a simulação para um grafo com $N = 100$, 360 dias e $m = 1000$ simulações. As probabilidades foram construídas de forma que os triângulos em que as arestas coincidem de forma invertida, vide Figura 23, a probabilidade do número de suscetíveis aumentar é zero, ou seja, $pji = 0$, pois não se considera um aumento do número de suscetíveis neste caso.

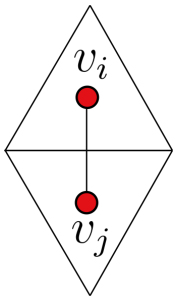


Figura 23 – Subtriângulos em posições invertidas do grafo. Nestes casos, a probabilidade de se passar do vértice v_j para v_i é zero. Fonte: esta pesquisa.

Na sequência de gráficos *boxplot* apresentada na Figura 24 visualiza-se graficamente que a dinâmica se dá de forma deveras semelhante ao real para todas as simulações geradas.

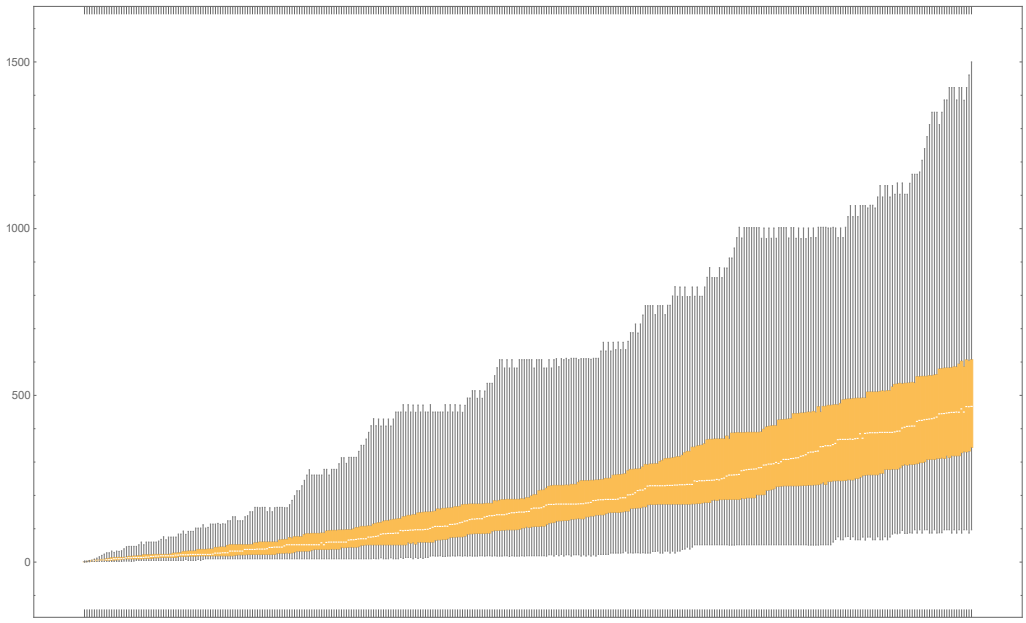


Figura 24 – Gráfico *boxplot* para $N = 100$ referente ao 1º ano (360 dias). No eixo horizontal tem-se dos dias da simulação e na vertical o índice dos vértices atingidos no conjunto das simulações para o dia correspondente. Fonte: esta pesquisa.

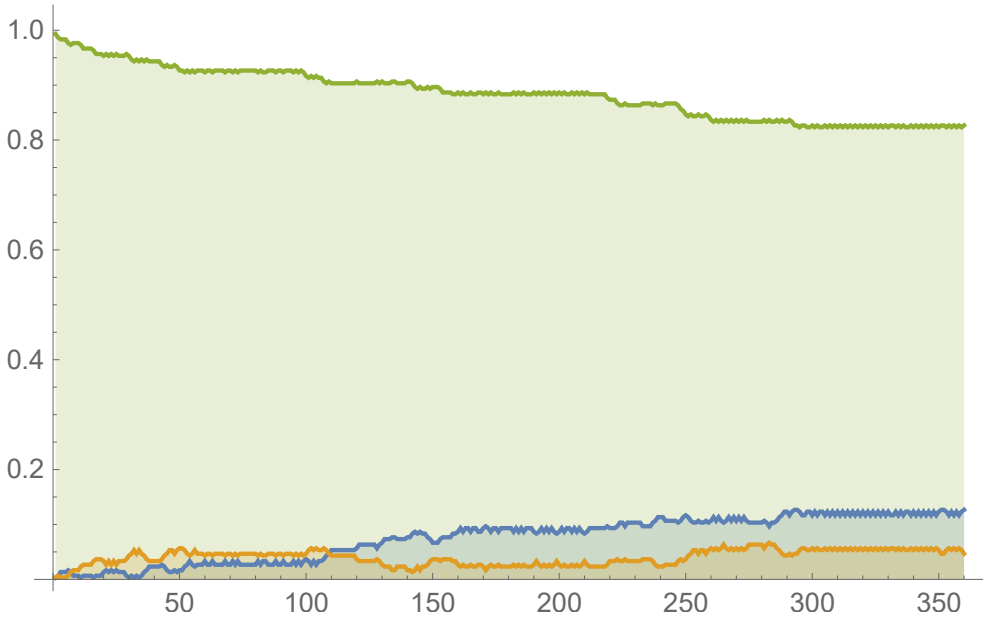


Figura 25 – Curvas das dinâmicas dos suscetíveis, infectados e recuperados para um grafo com $N = 100$. Curva verde: suscetíveis. Azul: infectados. Laranja: recuperados. Fonte: esta pesquisa. Fonte: esta pesquisa.

Para essa simulação é observado uma aderência maior ao modelo, utilizando as simulações geradas verifica-se que para o dia 360, a curva de suscetíveis, infectados e recuperados é de $78,3\%$, 4.3%, $17,4\%$, respectivamente.

Pela tabela resultante da *query* referente ao Anexo E para o dia 360, tem-se que:

6 Conclusões e outras considerações

6.1 Implementação do algoritmo para linguagem computacional e Dificuldades

Ao longo deste trabalho, foi proposto a construção de um grafo para discrição de uma dinâmica em, que representa a mudança de estados ao longo do tempo. Para isso, foram estabelecidas metas específicas, como a descrição de um sistema com três estados puros ou combinações de estados, a criação de um sistema de coordenadas para localizar o estado atual do sistema em relação aos três estados puros, a definição da dinâmica estocástica dos estados de um sistema específico, além da elaboração de um algoritmo que discretize o espaço de estados e gere um grafo e o modelo de difusão sobre o mesmo. As simulações realizadas a partir do modelo de discretização proposto mostram que é possível obter resultados muito próximos aos valores reais, mesmo quando as probabilidades são obtidas de maneira não fechada. A aplicabilidade e confiabilidade do algoritmo desenvolvido para sistemas complexos demonstra o potencial desta pesquisa em contribuir para o avanço do conhecimento em dinâmica de sistemas e fornecer ferramentas mais precisas e eficientes para análise e tomada de decisões em diferentes áreas.

A discretização de um sistema ternário em forma de um grafo, sugerido pelo modelo proposto foi testado e observado para diversos cenários. A primeira variação foi se a construção do Grafo $\mathcal{G}_N$ estava correta. Para isso foi verificado essa construção para parâmetros distintos.

A primeira variação foi dada pela quantidade de isolinhas, N, no qual é divido o grafo, essas isolinhas são as que ditam a quantidade de substriângulos que aparecem no grafo. Para esse primeiro parâmetro, inicialmente, foram utilizados valores variando de 5 em 5, $N = 5$, $N = 10, \cdots$ para que seja feito a verificação da veracidade dos resultados e construção do Grafo (comparando a saída do programa com o que seria esperado do algoritmo). Ainda na construção do Grafo, foi verificado se os vértices e seus respectivos vizinhos eram, de fato, os esperados do algoritmo. Após a escolha do

banco de dados utilizado nesse trabalho, a próxima etapa seria de fato discretizar esse sistema, sendo feita uma transformação do banco de dados real para o grafo $\mathcal{G}_N$ de forma a se tornar possível a implementação do modelo. O banco utilizado para esse trabalho foi o do Wesley Cota, disponibilizado no GitHub disponibilizado no link: <https://github.com/wcota/covid19br>. Utilizando a linguagem SQL em conjunto com o software open source PgAdmin, foi filtrado apenas os dados de interesse (total de casos, recuperados e data de registro). Como a informação de suscetíveis não era disposta de forma clara no bando de dados, foi feita uma rotina para estimar a população total de cada ano. Basicamente, dado os sensos do IBGE da quantidade de indivíduos no Brasil, para o ano de 2020, 2021, 2022 e 2023 e retirado os infectados e recuperados para cada dia, foi atribuído os valores para a população diárias de suscetíveis. Essa rotina é melhor exemplificada no Anexo E. De posse do grafo e do banco de dados tratado, foi de fato, feita a discretização do sistema real. Em suma, foi gerado uma lista contendo todos os dados disponíveis do Grafo, incluindo as coordenadas dos subtriângulos gerados pelas isolinhas do grafo nos quais os vértices são os baricentros desses triângulos. Dadas essas coordenadas e de posse do banco de dados real e tratado, o que é feito é uma comparação de coordenadas. Caso as porcentagens da população do banco real esteja dentro do subtriângulo do vértice, é retornado o baricentro daquele subtriângulo que é efetivamente o vértice v_ℓ do Grafo e assim se torna completa a discretização do banco de dados real. Logo têm-se o todas as informações do sistema discretizado, vértices, coordenadas baricêntricas, em qual região do grafo aquele dia se encontra e etc. O resultado final disso é a dinâmica real da doença dado o modelo proposto.

6.2 Aderência e potencialidade do modelo

Fundamentado pelas simulações computacionais geradas, o modelo sugerido para discretização de um sistema ternário é considerado adequado para modelagem de uma dinâmica. O modelo demonstra exatidão no que propõe pois mesmo dado probabilidades aleatórias e sem relação direta com um sistema específico, a dinâmica gerada é concordante com o esperado no mundo real, essa discretização de um sistema para um Grafo traz uma facilidade no tratamento de problemas antes tão complexos. Também é possível verificar que a construção do grafo se dá de forma eficiente pelos algoritmos listados, onde toda e qualquer informação do grafo pode ser gerada computacionalmente de forma simples. Ademais, não é aqui que sucumbe as possibilidades de aplicação desse modelo, caso as probabilidades de passagem de um vértice para outro fosse gerado pelas equações de Kolmogorov, seria visto uma dinâmica ainda mais semelhante com ao mundo real. Nesse trabalho foram restringidas as aplicações do modelo para o caso de uma dinâmica de

uma doença, mas vale salientar que a mesma modelagem pode ser aplicada para diversos sistemas ternários, dadas suas ressalvas.

6.3 Sobre a dinâmica sobre o grafo de estados mistos

Neste trabalho, pretendeu-se descrever a dinâmica estocástica dos estados de um sistema específico como um modelo de difusão, que envolve a função de distribuição acumulada condicional de um estado ser atingido em um determinado tempo futuro a partir de um ponto inicial arbitrário. Essa modelagem foi apresentada na Seção 4.5. Porém não se apresentou o uso de tais equações no cálculo das probabilidades de mudança de vértices para o grado $\mathcal{G}_N$. Pretende-se implementar futuramente um algoritmo que utilize a Equação 4.42 e suas consequências para indicar precisamente a dinâmica sobre um sistema ternário de estados.

Referências Bibliográficas

ANTONELLI, P.; STROBECK, C. The geometry and random drift i. stochastic distance and diffusion.
Advances in Applied Probability, v. 9, n. 2, p. 238–249, 1977.

BEAL, G. M.; BOHLEN, J. M. et al. **The diffusion process**. Iowa City, IA: Agricultural Experiment Station, Iowa State College, 1957.

BOLLOBÁS, B. **Modern graph theory**. New York, NY: Springer Science & Business Media, 1998. v. 184.

BONDY, J. A.; MURTY, U. S. R. et al. **Graph theory with applications**. New York, NY: Macmillan London Pub., 1976. v. 290.

CORMEN, T. H. et al. **Introduction to algorithms**. Cambridge, MA: MIT press, 2022.

DIESTEL, R. **Graph Theory**. New York: Springer, 2003.

ESSAM, J. W. Percolation theory.
Reports on progress in physics, IOP Publishing, v. 43, n. 7, p. 833, 1980.

IBE, O. **Markov processes for stochastic modeling**. Boston, MA: Elsevier/Academic Press, 2013.

JACOBS, M. H. **Diffusion processes**. New York, NY: Springer, 1935.

KRANTZ, S. G. **Real Analysis and Foundations**. 2^{nd} ed.. ed. Boca Raton: Chapman & Hall, 2005.

MAGALHÃES, M. N. **Probabilidade e variáveis aleatórias**. São Paulo, SP: Edusp, 2006.

MORGADO, A. C. de O. et al. Análise combinatória e probabilidade.
Sociedade Brasileira de Matemática, Rio de Janeiro, 1991.

PALS, D.; SPRY, P. Telluride mineralogy of the low-sulfidation epithermal emperor gold deposit, vatukoula, fiji.
Mineralogy and Petrology, Springer, v. 79, p. 285–307, 2003.

PORTENKO, N. I. **Generalized diffusion processes**. Providence, RI: American Mathematical Society, 1990. v. 83.

SAHIMI, M. **Applications of percolation theory**. New York, NY: Springer Nature, 2023. v. 213.

STAUFFER, D.; AHARONY, A. **Introduction to percolation theory**. Boca Raton, FL: CRC press, 2018.

WOLFRAM, S. Complex systems theory 1. In: **Emerging syntheses in science**. Boca Raton, Fl: CRC Press, 2018. p. 183–190.

ZUBEN, F. J. V.; CASTRO, L. N. de. Redes neurais.
Notas de aula, FEEC, Universidade Estadual de Campinas, Campinas, SP, 2003.

Anexos

ANEXO A – Código em Python da matriz de adjacência do grafo de estados mistos

```python
from math import floor, sqrt
import numpy as np
import csv
import random
from numpy.random import default_rng
''' [Algoritmo 01] Matriz de adjacência de $\mathcal{G}_N$ '''
# Inicia as variavéis
# N: número de níveis a serem considerados na partição do espaço de estados
N = 4
nivel_Lv1 = pos_v1 = 1
l = nivel_Lvl = pos_vl = 0
# Define a matriz indentidade $N^2 \times N^2$
matriz = np.eye(N**2, dtype=int)
matriz[1-1][2-1] = 1  # -1 é para evitar o inicio de contagem do python com 0
#Inicia arquivo pra guardar os vertices e seus respectivos vizinhos: i,j,k,l
vertices_vizinhos = open('vizinhos_N' + str(N) + '.csv','w', newline='', encoding='utf-8')
w = csv.writer(vertices_vizinhos)
for l in range(N**2):
    l = l + 1
    # Calcula o nível do vértice vl
    nivel_Lvl = max(floor(sqrt(4*l-4)), floor(sqrt(4*l-3)))
    if (nivel_Lvl % 2 == 0):
        # Calculando posição
        pos_vl = int(np.round((4*l - (nivel_Lvl)**2)/4))
        # Calculando vizinhos
        i = int(np.round((4*pos_vl + (nivel_Lvl - 1)**2 - 1)/4))
        j = int(np.round((4*pos_vl + (nivel_Lvl + 1)**2 - 1)/4))
        k = int(np.round((4*pos_vl + (nivel_Lvl + 1)**2 + 3)/4))
        # Adicionando elemento na matriz
        matriz[l-1, i-1] = 1
        matriz[l-1, j-1] = 1
        matriz[l-1, k-1] = 1
```

```
33      else:
34          # Calculando posição
35          pos_vl = int((4*l - (nivel_Lvl)**2 + 1)/4)
36          if nivel_Lvl != (2*N - 1):
37              if pos_vl == 1:
38                  # Calculando vizinhos
39                  i = int(np.round((4 + (nivel_Lvl - 1)**2)/4))
40                  j = int(np.round((4 + (nivel_Lvl + 1)**2)/4))
41                  k = 'NULL'
42                  # Adicionando elemento na matriz
43                  matriz[l-1, i-1] = 1
44                  matriz[l-1, j-1] = 1
45              else:
46                  if int(pos_vl == ((nivel_Lvl + 1)/2)):
47                      # Calculando vizinhos
48                      i = int(np.round((2*(nivel_Lvl - 1) + (nivel_Lvl -
                            1)**2)/4))
49                      j = int(np.round((2*(nivel_Lvl + 1) + (nivel_Lvl +
                            1)**2)/4))
50                      k = 'NULL'
51                      # Adicionando elementos na matriz
52                      matriz[l-1, i-1] = 1
53                      matriz[l-1, j-1] = 1
54                  else:
55                      # Calculando vizinhos
56                      i = int(np.round((4*(pos_vl - 1) + (nivel_Lvl - 1)
                            **2)/4))
57                      j = int(np.round((4*pos_vl + (nivel_Lvl - 1)**2)/4))
58                      k = int(np.round((4*pos_vl + (nivel_Lvl + 1)**2)/4))
59                      # Adicionando elementos na matriz
60                      matriz[l-1, i-1] = 1
61                      matriz[l-1, j-1] = 1
62                      matriz[l-1, k-1] = 1
63          else:
64              if pos_vl == 1:
65                  # Calculando vizinhos
66                  i = int(np.round((4 + (nivel_Lvl - 1)**2)/4))
67                  j = 'NULL'
68                  k = 'NULL'
69                  # Adicionando elementos na matriz
70                  matriz[l-1, i-1] = 1
71              else:
72                  if int(pos_vl == (nivel_Lvl + 1)/2):
73                      # Calculando vizinhos
```

```
74                      i = int(np.round((2*(nivel_Lvl - 1) + (nivel_Lvl -
                            1)**2)/4))
75                      j = 'NULL'
76                      k = 'NULL'
77                      # Adicionando elementos na matriz
78                      matriz[l-1, i-1] = 1
79                  else:
80                      # Calculando vizinhos
81                      i = int(np.round((4*(pos_vl - 1) + (nivel_Lvl - 1)
                            **2)/4))
82                      j = int(np.round((4*pos_vl + (nivel_Lvl - 1)**2)/4))
83                      k = 'NULL'
84                      # Adicionando elementos na matriz
85                      matriz[l, i-1] = 1
86                      matriz[l, j-1] = 1
87      w.writerow([i,j,k,l])
88  vertices_vizinhos.close()
```

Listing A.1 – Matriz de adjacência do Grafo $\mathcal{G}_N$ na linguagem Python

ANEXO B – Código em Python para cálculo das coordenadas dos vértices do grafo de estados

```
'''[Algoritmo 02] Coordenadas baricêntricas de cada vértice v_l de G_N
    Usou-se o int(np.round()) pra evitar que os índices i,j,k retorne
       valores float
    Utiliza os dados do algoritmo 01 [Matriz de adjacência]'''

if nivel_Lvl % 2 == 0:
    si0 = round(1 - (3*nivel_Lvl + 1)/(6*N), 8)
    si1 = round((3*pos_vl - 2)/(3*N), 8)
    si2 = round((3*nivel_Lvl - 6*pos_vl + 5)/(6*N), 8)
else:
    si0 = round(1 - (3*nivel_Lvl - 2)/(6*N), 8)
    si1 = round((3*pos_vl - 1)/(3*N), 8)
    si2 = round((3*nivel_Lvl - 6*pos_vl + 4)/(6*N), 8)
```

Listing B.1 – Coordenadas baricêntricas do vértice v_ℓ

ANEXO C – Código em Python da dinâmica da simulação do grafo de estados mistos

```
from math import floor, sqrt
import numpy as np
import csv
import random
from numpy.random import default_rng
import pandas as pd

df = pd.read_csv('vertices_N' + str(N) '.csv')
dia = 1
N = 100

dados_grafo = open('simulacao' + str(tempo) + '.csv',
                   'w', newline='', encoding='utf-8')
w = csv.writer(dados_grafo)
v_atual = 1
trajetoria = v_atual
# trajetoria = []
# trajetoria = [[0 for x in range(N)] for x in range(tempo)]
trajetoria = np.array([[], []])
simulacoes = 1000

for m in range(1, simulacoes):
    v_atual = 1
    while dia <= tempo:
        dia += 1
        u = random.random()
        if u <= df.iat[v_atual, 7]:  # pll
            sn = df.iat[v_atual, 0]  # l
            trajetoria[(dia), (simulacoes)].append(sn)

        elif (df.iat[v_atual, 7] < u <= (df.iat[v_atual, 7] + df.iat[
            v_atual, 9])):
            sn = df.iat[v_atual, 1]  # i
            trajetoria[(dia), (simulacoes)].append(sn)

        elif ((df.iat[v_atual, 7] + df.iat[v_atual, 9]) < u <= (df.iat[
```

```
            v_atual, 7] + df.iat[v_atual, 9] + df.iat[v_atual, 10])):
36              sn = df.iat[v_atual, 2]  # j
37              trajetoria[(dia), (simulacoes)].append(sn)
38
39          elif ((df.iat[v_atual, 7] + df.iat[v_atual, 9] + df.iat[v_atual,
                10]) < u <= 1):
40              sn = df.iat[v_atual, 3]  # k
41              trajetoria[(dia), (simulacoes-5)].append(sn)
42          v_atual = trajetoria[(dia), (simulacoes)]
43      w.writerow([trajetoria[m, dia]])
44  dados_grafo.close()
```

Listing C.1 – Simulação da dinâmica da caminhada dos vértices v_ℓ

ANEXO D – DDL: Importando os dados para o pgAdmin4

```
CREATE TABLE covid_br (
  ID serial PRIMARY KEY,
  epi_week NUMERIC(10),
  date DATE,
  country VARCHAR(50),
  state VARCHAR(50),
  city VARCHAR(50),
  newDeaths NUMERIC(100),
  deaths  NUMERIC(100),
  newCases NUMERIC(100),
  totalCases  NUMERIC(100),
  deathsMS NUMERIC(100),
  totalCasesMS NUMERIC(100),
  deaths_per_100k_inhabitants NUMERIC(100),
  totalCases_per_100k_inhabitants NUMERIC(100),
  deaths_by_totalCases NUMERIC(100),
  recovered NUMERIC(100),
  suspects NUMERIC(100),
  tests NUMERIC(100),
  tests_per_100k_inhabitants NUMERIC(100),
  vaccinated  NUMERIC(100),
  vaccinated_per_100_inhabitants NUMERIC(100),
  vaccinated_second NUMERIC(100),
  vaccinated_second_per_100_inhabitants NUMERIC(100),
  vaccinated_single NUMERIC(100),
  vaccinated_single_per_100_inhabitants NUMERIC(100),
  vaccinated_third NUMERIC(100),
  vaccinated_third_per_100_inhabitants NUMERIC(100)
);

COPY covid_br() FROM '[endereço onde está seu arquivo .csv]' DELIMITER '
    ,' CSV HEADER;
```

Listing D.1 – PostgreSQL: Script de inserção do banco de dados no SQL

ANEXO E – DQL: Consultas no banco

```
WITH coord_covid AS(
  SELECT
    date AS "data_registro",
    CASE
      WHEN EXTRACT(year from date) = 2020 THEN ((213200000 - totalcases
         - deaths)/213200000)
      WHEN EXTRACT(year from date) = 2021 THEN ((214300000 - totalcases
          - deaths)/214300000)
      WHEN EXTRACT(year from date) = 2022 THEN ((207750291  - totalcases
          - deaths)/207750291)
      END "n_suscetiveis",
    CASE
      WHEN EXTRACT(year from date) = 2020 THEN ((totalcases + suspects)
         /213200000)
      WHEN EXTRACT(year from date) = 2021 THEN ((totalcases + suspects)
         /214300000)
      WHEN EXTRACT(year from date) = 2022 THEN ((totalcases + suspects)
         /207750291)
      END "n_infectados",
    CASE
      WHEN EXTRACT(year from date) = 2020 THEN ((recovered)/213200000)
      WHEN EXTRACT(year from date) = 2021 THEN ((recovered)/214300000)
      WHEN EXTRACT(year from date) = 2022 THEN ((recovered)/207750291)
      END "n_recuperados"
  FROM covid_br
    WHERE state = 'TOTAL'
)
SELECT * FROM coord_covid
```

Listing E.1 – PostgreSQL: Tratamento de dados do banco do COVID-19 no Brasil

```
WITH coord_covid AS(
  SELECT
    date AS "data_registro",
    CASE
      WHEN EXTRACT(year from date) = 2020 THEN ((213200000 - totalcases
         - deaths)/213200000)
      WHEN EXTRACT(year from date) = 2021 THEN ((214300000 - totalcases
          - deaths)/214300000)
```

```
7       WHEN EXTRACT(year from date) = 2022 THEN ((207750291  - totalcases
            - deaths)/207750291)
8       END "n_suscetiveis",
9     CASE
10      WHEN EXTRACT(year from date) = 2020 THEN ((totalcases + suspects)
           /213200000)
11      WHEN EXTRACT(year from date) = 2021 THEN ((totalcases + suspects)
           /214300000)
12      WHEN EXTRACT(year from date) = 2022 THEN ((totalcases + suspects)
           /207750291)
13      END "n_infectados",
14    CASE
15      WHEN EXTRACT(year from date) = 2020 THEN ((recovered)/213200000)
16      WHEN EXTRACT(year from date) = 2021 THEN ((recovered)/214300000)
17      WHEN EXTRACT(year from date) = 2022 THEN ((recovered)/207750291)
18      END "n_recuperados"
19  FROM covid_br
20    WHERE state = 'TOTAL'
21 )
22 SELECT
23   *
24 FROM(
25   SELECT
26     CASE
27       WHEN (nivel_Lvl % 2 <> 0) THEN 1
28         WHEN (c."n_suscetiveis" <= g.w0_x) THEN 1
29             WHEN (c."n_infectados" >= g.w0_y) AND (c."n_infectados" <= g
                  .w1_y) THEN 1
30               WHEN (c."n_recuperados" >= g.w1_z) AND (c."n_recuperados"
                    <= g.w2_z) THEN 1
31     END AS vertice
32   FROM coord_covid c
33     JOIN grafon10_4 g ON c."n_suscetiveis" >= g.w2_x
34   UNION
35   SELECT
36     CASE
37       WHEN (nivel_Lvl % 2 = 0) THEN 1
38         WHEN (c."n_suscetiveis" <= g.w0_x) THEN 1
39             WHEN (c."n_infectados" >= g.w0_y) AND (c."n_infectados" <= g
                  .w1_y) THEN 1
40               WHEN (c."n_recuperados" >= g.w1_z) AND (c."n_recuperados"
                    <= g.w2_z) THEN 1
41     END AS vertice
42   FROM coord_covid c
43     JOIN grafon10_4 g ON c."n_suscetiveis" >= g.w1_x
```

```
44   )x
45 ORDER BY x.vertice ASC
```

Listing E.2 – PostgreSQL: Dinâmica da simulação