

# Timelines e Controle de Epidemias

UFRJ – Ciência da Computação, AD 2020/1

September 26, 2020

## Abstract

O objetivo deste trabalho de simulação é introduzir conceitos básicos de um simulador de eventos discretos, e experimentar diferentes fatores que impactam na disseminação de fake news em mídias sociais.

## 1 Introdução

Em redes de mídias sociais como Facebook, Twitter, Instagram ou LinkedIn, cada usuário possui sua própria *timeline* (linha do tempo), que é preenchida por postagens do próprio usuário, pelas postagens dos amigos (usuários seguidos diretamente), ou ainda de usuários indiretamente seguidos (seguido por algum amigo), com base em algum algoritmo de recomendação da plataforma.

O objetivo deste trabalho é implementar um simulador que nos permita correlacionar a dinâmica de conteúdo em *timelines* [HAM<sup>+</sup>19, GBV19] e a dinâmica de epidemias em redes [PZE<sup>+</sup>13, PZE<sup>+</sup>14], e verificar a possibilidade de controle de visibilidade de conteúdo em mídias sociais.

## 2 Conceitos Básicos

### 2.1 Conteúdos em Timelines

Em uma timeline, temos posts que são alimentados por fontes externas e por amigos. A timeline pode conter um ou mais fake news. Vamos considerar duas variantes do problema:

- timeline funcionando conforme first-in first-out (FIFO), na qual quando chega um novo post na timeline, o post mais de baixo é removido, os outros posts descem uma posição, e o novo post entra no topo da timeline;
- timeline funcionando conforme estratégia aleatória (RND), na qual quando chega um novo post na timeline, um dos posts lá existentes é removido de forma uniforme e aleatória.

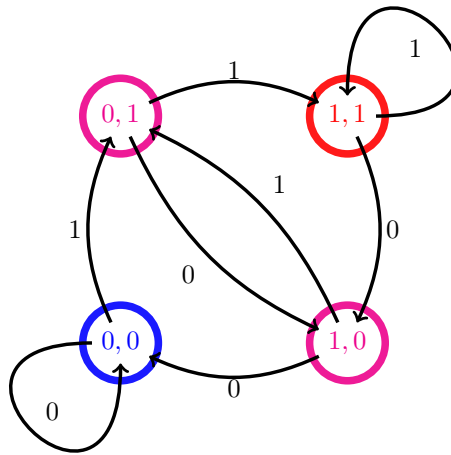


Figure 1: Exemplo de uma máquina de estados capturando o estado de um único nó, operando sob política FIFO: os elementos 0 e 1 sobre as arestas representam o tipo de conteúdo recebido. Cada aresta representa uma transição, em função do tipo de conteúdo recebido. Os quatro estados são: (0,0) corresponde ao estado em que o nó não possui fake news, (0,1) ao estado que possui 1 fake news no topo da timeline, (1,0) ao estado que possui 1 fake news na parte de baixo da timeline, e (1,1) ao estado que possui fake news nas duas posições da timeline.

## 2.2 Epidemias em timelines

Consideramos uma população finita de  $N$  usuários, conectados em uma clique (todos os usuários estão conectados com todos os outros).

Cada usuário pode estar em um dentre dois estados:

1. **suscetível (S):** um nó no estado suscetível pode tornar-se infectado seja por conta de infecções exógenas (agente infectante fora da rede de amizade, e.g.: um post de um jornal ou revista) ou endógenas (agente infectante participante da rede, e.g.: um vizinho contaminado que posta algo na timeline)
2. **infectado (I):** um nó infectado pode infectar seus vizinhos. Um nó infectado torna-se limpo quando sua timeline não possui mais nenhuma fake news.

Seja  $K$  o número de posições na timeline. A seguir, assumimos que  $K = 2$ , ou seja, estamos interessados apenas nas 2 primeiras posições da timeline.

Assuma que infecções exógenas (posts com fake news exógenos) chegam a cada nó segundo um fluxo Poisson com taxa  $\lambda_1$ . Posts bons chegam de forma exógena a cada nó segundo fluxo Poisson com taxa  $\lambda_0$ .

Assuma que cada nó infectado infecta cada um de seus vizinhos segundo um fluxo Poisson com taxa  $\mu_1 f_1(a)$ , onde  $a$  é o número de posts fake na timeline do usuário infectante.

Além disso, assuma que cada nó envia boas notícias a cada um de seus vizinhos segundo um fluxo Poisson com taxa  $\mu_0 f_0(K - a)$ , onde  $K - a$  é o número de bons posts na timeline do usuário em questão transmitindo conteúdo para seus vizinhos.

Em geral, assumimos que  $f_1(1) = f_0(1) = 1$ .

A figura 2 apresenta um modelo de transição de estados para um nó.

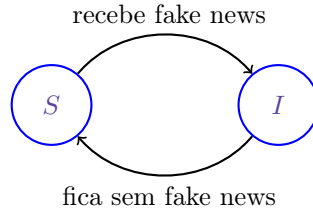


Figure 2: Diagrama de estados para o modelo epidêmico SIS.

## 3 Cadeias de Markov

As duas figuras a seguir ilustram as cadeias de Markov correspondentes ao sistema FIFO e RND, respectivamente. Note que nessas cadeias assumimos  $\lambda_0 = \lambda_1 = 0$ .

$\mu_0$	Taxa de transmissão de boas notícias, por post, por usuário
$\mu_1$	Taxa de transmissão de más notícias (fake news), por post, por usuário
$S$	Estado suscetível
$I$	Estado infectado
$N$	Tamanho da população
$K$	tamanho da timeline (assuma 2, caso não seja dito nada contrário)
$\lambda_1$	Taxa de infecção exógena por nó
$\lambda_0$	Taxa de chegada de bons posts exógena por nó
$p(t)$	probabilidade de haver fake news na timeline de um usuário, no tempo $t$
$n_0$	número de usuários sem nenhuma fake news na timeline
$n_1$	número de usuários com uma fake news na timeline
$n_2$	número de usuários com duas fake news na timeline
$n_{0,0}$	número de usuários sem fake news em nenhuma das duas posições da timeline
$n_{1,0}$	número de usuários com fake news na segunda posição da timeline, mas com o topo (primeira posição) limpo
$n_{0,1}$	número de usuários com fake news no topo da timeline, mas com a parte de baixo limpa
$n_{1,1}$	número de usuários com fake news no topo e na parte de baixo da timeline
$f_1(a)\mu_1$	taxa com que usuário com $a$ fake news infecta cada um de seus vizinhos
$f_0(K-a)\mu_0$	taxa com que usuário com $K-a$ posts limpos envia boas notícias para cada um de seus vizinhos

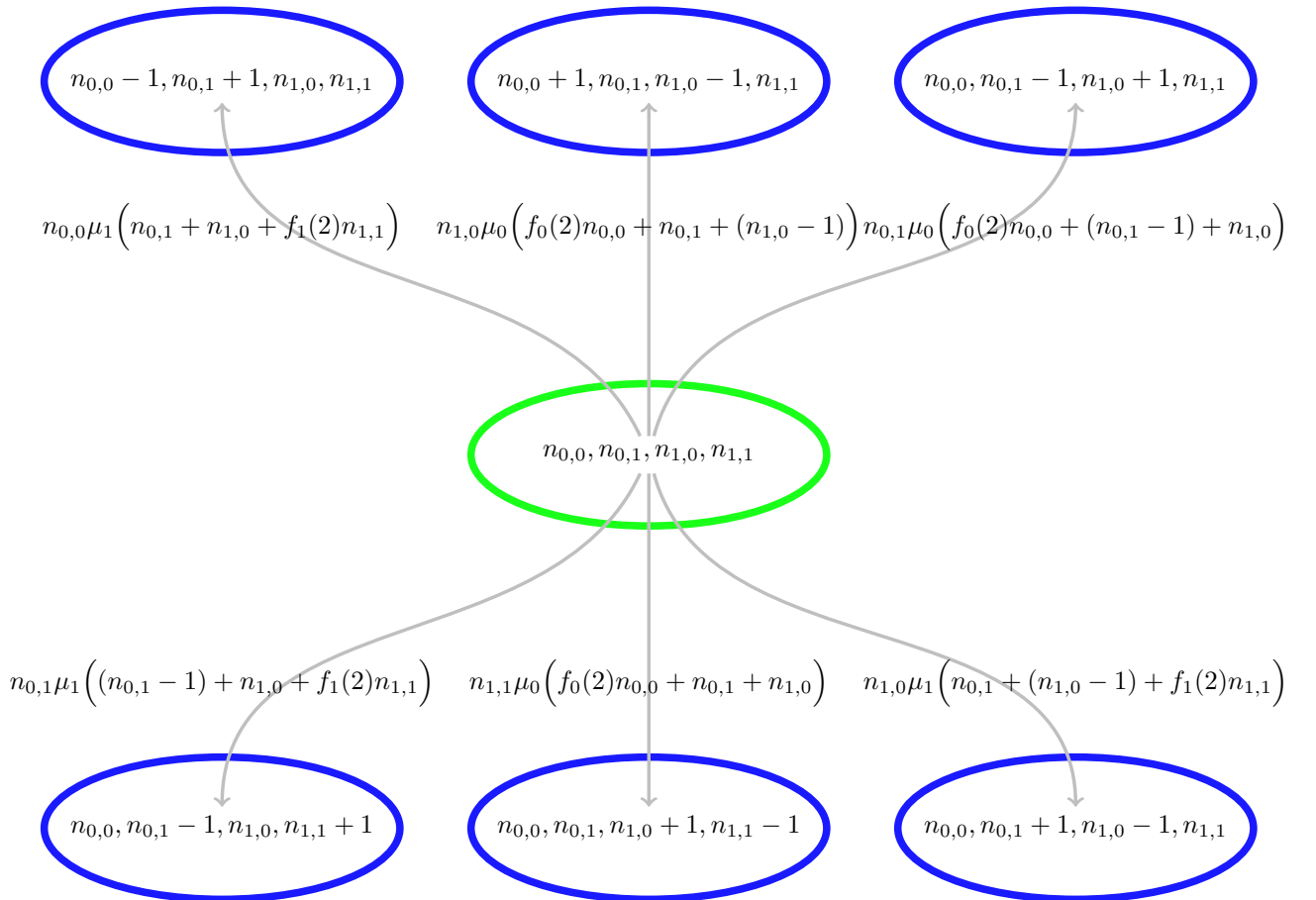


Figure 3: Cadeia de Markov representando estado do sistema: caso FIFO

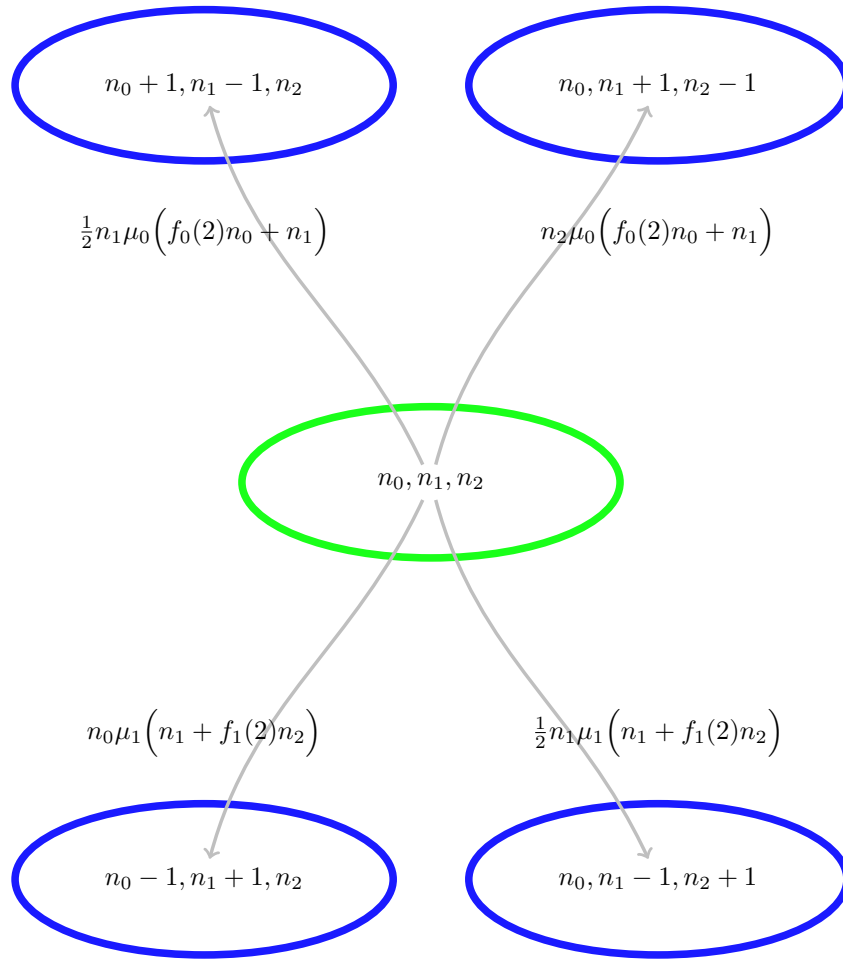


Figure 4: Cadeia de Markov representando estado do sistema: caso RND

## 4 Exemplo ilustrativo

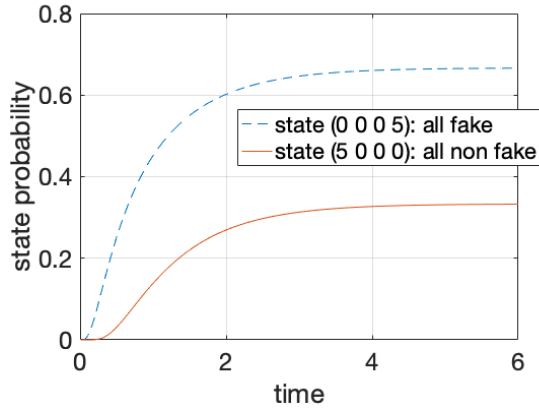
As Figuras 5-6 mostram exemplos ilustrativos de como a população evolui com o tempo. Usando Matlab, calculamos a solução do problema, e capturamos a evolução da população ao longo do tempo. Quando consideramos  $\lambda_0 = \lambda_1 = 0$ , o sistema leva aproximadamente 4 unidades de tempo até atingir estado aborrvante. Já quando consideramos  $\lambda_0 = 0.1$  e  $\lambda_1 = 0.2$ , o sistema fica no estado em que todos os nós estão infectados em torno de 50% do tempo.

Será que você consegue reproduzir estes comportamentos em uma simulação? Esse é um dos objetivos deste trabalho.

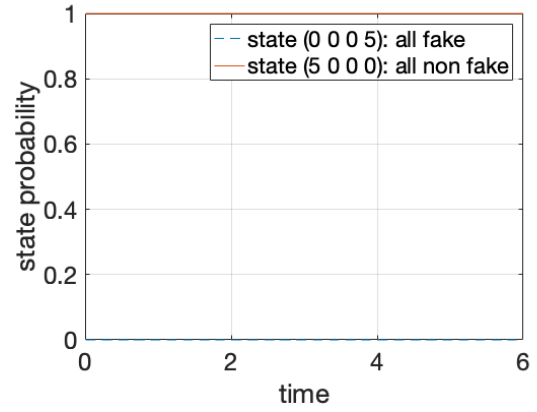
1. **Primeira parte:** fazer simulações, construindo um simulador de eventos discretos, produzindo resultados com intervalos de confiança, capturando o comportamento destes dois sistemas, nos casos FIFO e RND, lhe confere metade da nota.
2. **Segunda parte:** a outra metade envolve criatividade: extrapolar esses exemplos, e responder algumas das perguntas em aberto apontadas na próxima seção.

Atenção! Um dos propósitos deste trabalho é construir um simulador de eventos discretos. Isso significa que você precisa escrever um programa no qual, necessariamente, no loop principal, você remova eventos de uma fila de eventos, trate esses eventos, e possivelmente insira novos eventos na fila de eventos.

- o seu programa necessariamente precisa ter uma fila de eventos, idealmente implementada como uma fila com prioridades (todas as linguagens oferecem essa estrutura)
- o seu programa precisa, no loop principal, remover eventos da fila de eventos, tratar tais eventos, e inserir novos eventos na fila caso necessário



(a)  $\lambda_0 = \lambda_1 = 0$ , estado inicial=(0 5 0 0)



(b)  $\lambda_0 = \lambda_1 = 0$ , estado inicial=(5 0 0 0)

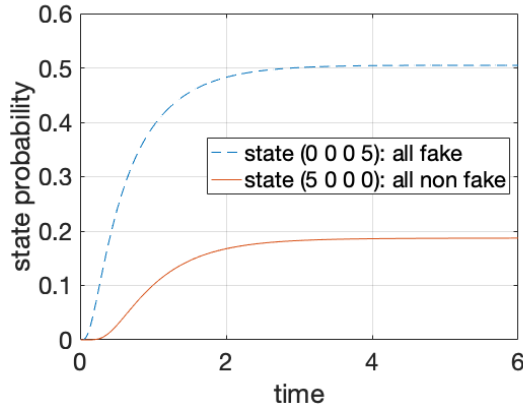
Figure 5: Sistema transiente: dependendo do estado inicial, o estado final do sistema irá mudar. Em outras palavras, depois de um tempo longo, o estado final do sistema depende do estado inicial. Há dois estados absorventes: o estado em que todos os nós só tem fake news, e o estado em que todos só tem boas notícias. (a) O estado inicial (0 5 0 0) corresponde a 5 nós inicialmente terem uma fake news no topo da timeline. O sistema no fim das contas tem uma chance grande de acabar tendo todos os nós com a fake news. (b) O estado inicial (5 0 0 0) corresponde a 5 nós inicialmente terem boas notícias. O sistema fica aí para sempre.

#### 4.1 Sistema transiente: tempo até extinção da epidemia

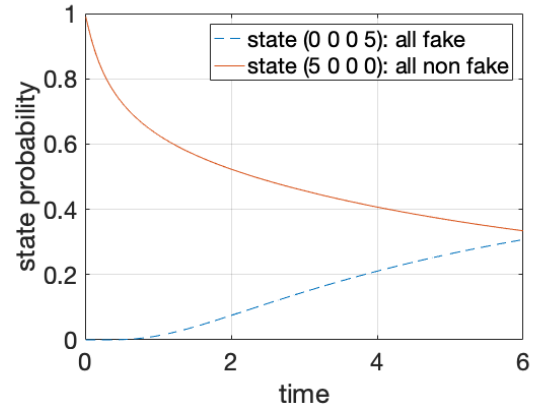
Primeiro consideramos o caso em que não há chegadas de mensagens exógenas. Ou seja, tudo que é trocado na rede são mensagens entre os nós. Note que quando iniciamos o sistema no estado em que todos os nós tem uma notícia fake news no topo da timeline, existe uma chance grande de o sistema acabar no estado em que todos os usuários tem fake news em todas as posições.

**Pergunta 1:** por que a solução do sistema não é simétrica? Ou seja, por que no fim das contas não temos que com probabilidade 50% terminamos no estado all fake, e com 50% no estado all non fake? O que está quebrando a simetria?

**Pergunta 2:** como se comporta a simulação para diferentes condições de estado inicial?



(a)  $\lambda_0 = 0.1, \lambda_1 = 0.2$ , estado inicial=(0 5 0 0)



(b)  $\lambda_0 = 0.1, \lambda_1 = 0.2$ , estado inicial=(5 0 0 0)

Figure 6: Sistema estacionário: o sistema está sujeito a constantes injeções de boas notícias e fake news exógenas. O sistema deveria depender das condições iniciais nesse caso? As figuras acima parecem sugerir que o sistema está dependendo da condição inicial. Como resolver esse mistério?

## 4.2 Sistema estacionário: solução estacionária

Agora consideramos o caso em que há chegadas de mensagens exógenas. Ou seja, além do que é trocado na rede entre os nós, também chegam mensagens de fora. Note que quando iniciamos o sistema no estado em que nenhum nó tem notícia fake na timeline, ainda chegam notícias fake de fora, então o sistema continua evoluindo a partir daí (compare com o caso em que não havia chegadas exógenas, em que o sistema acabava no estado inicial para sempre).

**Pergunta 1:** por que a solução do sistema nas figuras parece depender da condição inicial? A solução não deveria ser independente da condição inicial? Qual o mistério?

**Pergunta 2:** como se comporta a simulação para diferentes condições de estado inicial?



## 5 Questões

Procure rodar cada cenário considerado no mínimo 10 vezes (ou seja, execute pelo menos 10 rodadas), e apresente no seu relatório, para cada cenário, e para cada métrica considerada, o mínimo, o máximo, e a média obtida ao longo destas 10 rodadas. É imprescindível que se apresente também intervalos de confiança. Uma das métricas de interesse é o número médio de nós infectados. Outras possíveis métricas de interesse incluem a probabilidade de um nó marcado estar infectado, e o tempo médio até a população inteira ficar sem infecção ou ficar toda infectada.

Note que no caso do sistema com estados absorventes, esse problema se parece, em parte, com a ruína do apostador: temos dois estados absorventes. Um dos estados corresponde à população inteira estar curada, e o outro à população inteira estar sem fake news.

Considere, por exemplo, o número médio de nós infectados. Para cada uma das 10 rodadas você irá obter uma média. Sejam  $I_1, I_2, \dots, I_{10}$  essas médias. Apresentem o mínimo, o máximo, e a média desses 10 números. Mostre também o intervalo de confiança deles.

### 5.1 Parte 1: Primeiros Passos

O seu primeiro objetivo no trabalho é reproduzir os resultados das figuras acima, tanto para o caso FIFO quanto RND, via simulação e via cadeias de Markov. Simule o sistema, inclua intervalos de confiança, e verifique se, de fato, os valores obtidos via Cadeias de Markov estão dentro dos intervalos de confiança da simulação.

### 5.2 Parte 2: Usando a Criatividade

O segundo objetivo do trabalho é fazer variações no problema e indicar suas descobertas. Eis aqui algumas ideias, mas você pode usar sua criatividade para inventar outras perguntas e responde-las.

- topologia: nos exemplos acima consideramos uma topologia completamente conectada. O que ocorreria se a topologia fosse outra? Por exemplo, considerar uma estrela ao invés de uma clique?
- taxas: qual o efeito das taxas de infecção exógena e endógena?
- estado inicial: qual o efeito do estado inicial da simulação? O que ocorre se inicialmente tivermos mais ou menos fake news na rede?
- inserção de boas notícias versus filtragem de fake news: o que é mais eficiente? Injetar mais boas notícias? Ou tentar filtrar as fake news?
- distribuição dos tempos entre eventos: até então assumimos que os tempos entre eventos são exponenciais. Você consegue simular outros casos? Por exemplo, consegue simular os casos em que alguns dos tempos entre eventos são determinísticos?

Você pode usar-se de sua criatividade para variar o processo de epidemia, e ver os efeitos da variação em cima dos resultados.

## 6 Implementação

Você pode implementar o programa na linguagem que julgar mais adequada.

Note que ao implementar o simulador você irá se deparar com várias decisões de projeto. Documente e justifique suas decisões! Inclusive, você pode querer começar implementando um simulador simples, por exemplo, fortemente baseado nas cadeias de Markov apresentadas nesse enunciado, e aos poucos ir flexibilizando o simulador para poder levar em conta casos não contemplados pelas cadeias de Markov. Alternativamente, você pode preferir inicialmente já ir implementando um simulador mais genérico, conforme indicado a seguir.

Uma das várias maneiras simples de implementar a simulação consiste em associar a cada nó da rede (indivíduo) um identificador. Quando esse indivíduo está infectado, escalona-se eventos de infecção dos vizinhos dele. Quando ele fica livre de fake news, todos os eventos de infecção dele para os vizinhos são removidos. Em todo caso também escalona-se eventos de transmissão de boas mensagens do nó para os vizinhos, sempre com taxas ajustadas de acordo com o número de boas notícias e fake news na timeline do usuário.

1. evento de infectar vizinho com fake news: esse evento insere uma fake news em um vizinho. Se o vizinho já estiver com todas as posições da timeline ocupadas com fake news, ele continuará com todas as posições ocupadas com fake news. Lembre que, no caso mais simples, o tempo entre infecções é dado por uma variável aleatória exponencial, com média que depende do número de fake news do nó contaminante;
2. evento de transmitir ao vizinho com boa notícia: similar ao evento acima, mas agora para boas notícias;
3. infectar-se exogenamente: esse evento corresponde a entrada de uma nova mensagem fake news na timeline de um usuário, independente do estado dos vizinhos. Ela está associada a um timer, que toca depois de um tempo exponencialmente distribuído com média  $1/\lambda_1$ ;
4. receber boa notícia exogenamente: esse evento corresponde a entrada de uma nova mensagem boa na timeline de um usuário, independente do estado dos vizinhos. Ela está associada a um timer, que toca depois de um tempo exponencialmente distribuído com média  $1/\lambda_0$ .

Para implementar timers exponenciais, você pode proceder da seguinte forma. Primeiro, colha amostras uniformes usando o gerador de números pseudo-aleatórios do seu sistema (e.g., `random.random()` em Python). Em seguida, aplique uma função que converte a amostra uniforme  $u$  em uma amostra exponencial. Se a taxa for  $\lambda$ , por exemplo, aplique na amostra  $u$  a função  $-\log(u)/\lambda$ , que irá retornar uma amostra  $a$ . Alternativamente, em Python, você pode chamar direto `numpy.random.exponential()`. Em seguida, é só incluir o evento na lista de eventos.

## 7 O que incluir no relatório

O seu relatório deve conter, no mínimo, as seguintes seções

1. introdução: visão geral sobre o trabalho
2. parte 1: simulações básicas (metade da nota)
  - (a) simulações básicas caso FIFO: mostrar com intervalos de confiança que o seu simulador está de acordo com as cadeias de Markov apresentadas nesse enunciado
  - (b) simulações básicas caso RND: mostrar com intervalos de confiança que o seu simulador está de acordo com as variantes das cadeias de Markov apresentadas nesse enunciado, que vocês irão implementar
3. parte 2: simulações criativas (metade da nota)
  - (a) pense em questões que julgue interessante sobre propagação de epidemias de fake news
  - (b) simule sistemas, variantes do sistema acima, para responder suas questões – atenção! as simulações precisam, necessariamente, envolver timelines
4. desafios e decisões de projeto: liste os desafios encontrados ao realizar o projeto, e as decisões de projeto tomadas
5. referências: indique todos os sites e livros consultados

## 8 Nota

A sua nota será proporcional a

1. qualidade dos seus resultados,
2. criatividade na variação de parâmetros e cenários,
3. profundidade da discussão de resultados,
4. detalhamento das abordagens usadas para verificação da corretude do seu simulador, incluindo gráficos e números que comprovem que seu simulador está correto em vários casos base simples

## 9 O que entregar?

1. link para github ou algum outro repositório com código
2. relatório em PDF

Atenção! O relatório é extremamente importante, e é baseado nele (junto com o código) que iremos dar a nota. Não deixe para escrever o relatório na última hora. Planeje-se.

## 10 Dúvidas

Dúvidas devem ser direcionadas ao professor, via email, de preferência com cópia para o restante da turma.

## 11 Material disponível no google drive

Vide arquivos em <https://tinyurl.com/simulepi>

As Figuras 5-6 foram obtidas usando o código acima. Na Figura 5 consideramos  $\lambda_0=0$ ;  $\lambda_1=0$ ; e na Figura 6 consideramos  $\lambda_0=0.1$ ;  $\lambda_1=0.2$ . Nas figuras 5(a) e 6(a) e consideramos o parâmetro  $\text{inistate}=36$ ; e nas outras  $\text{inistate}=1$ . Fora isso, as figuras são obtidas usando exatamente o mesmo código. As mudanças desses parâmetros podem ser feitas no início do código, logo após as linhas “parameters to be configured”.

O código não está extremamente elegante, mas ele funciona. Qualquer dúvida, podem perguntar. Se acharem o código muito confuso, vocês podem também fazer o próprio código do zero, sem olhar para esse código fornecido.

É importante ressaltar que o código fornecido é para montar e resolver uma cadeia de Markov. O código não é um simulador! O simulador vocês irão precisar implementar do zero.

## 12 Prazos de entrega

- rascunho inicial: 12 de outubro
- trabalho semi-pronto: 26 de outubro
- trabalho final e apresentações: 16 de novembro

## References

- [GBV19] Anastasios Giovanidis, Bruno Baynat, and Antoine Vendeville. Performance analysis of online social platforms. *INFOCOM*, 2019. arXiv preprint arXiv:1902.07187.
- [HAM<sup>+</sup>19] Eduardo Hargreaves, Claudio Agosti, Daniel Menasché, Giovanni Neglia, Alexandre Reiffers-Masson, and Eitan Altman. Fairness in online social network timelines: Measurements, models and mechanism design. *Performance Evaluation*, 129:15–39, feb 2019.
- [PZE<sup>+</sup>13] Victor M Preciado, Michael Zargham, Chinwendu Enyioha, Ali Jadbabaie, and George Pappas. Optimal vaccine allocation to control epidemic outbreaks in arbitrary networks. In *52nd IEEE conference on decision and control*, pages 7486–7491. IEEE, 2013.
- [PZE<sup>+</sup>14] Victor M Preciado, Michael Zargham, Chinwendu Enyioha, Ali Jadbabaie, and George Pappas. Optimal resource allocation for network protection: A geometric programming approach. *IEEE Transactions on Control of Network Systems*, 1(1):99–108, 2014.