Uma introdução aos grafos de infecção

Priscila Cardoso Calegari

Departamento de Informática e Estatística

Abril de 2024

UNIVERSIDADE FEDERAL
DE SANTA CATARINA

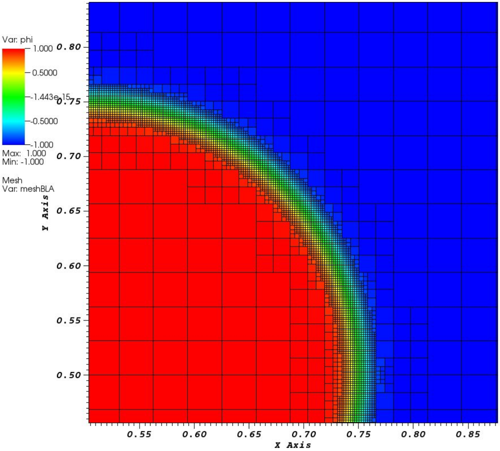
Sobre o que vamos conversar

- Motivação.
- Fundamentos.
- Modelo matemático.
- Métodos Numéricos.
- Grafos.
- Experimentos numéricos.

Motivação

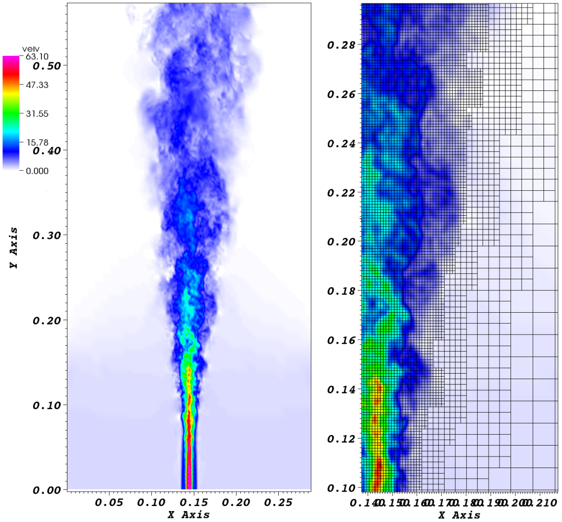
- A área da Computação Científica se refere a um conjunto de algoritmos, técnicas e teorias usadas para resolver problemas da ciência e da engenharia com o auxílio de um computador.
- Problemas de aplicação: Engenharia aerospacial, Engenharia Mecânica, Engenharia Elétrica, Engenharia Química, Astronomia, Meteorologia, Biologia, Ciências Médicas, ...

Motivação



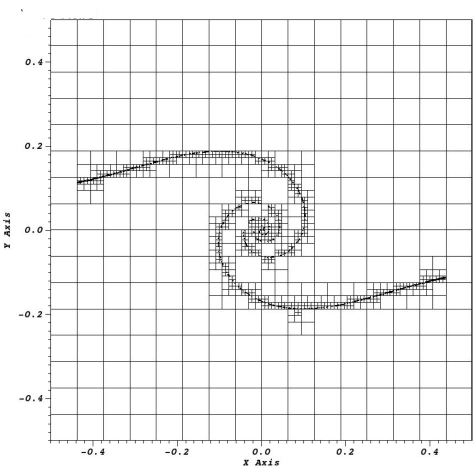
Var: phi
1.000
0.5000
-1.443e-15
-0.5000
-1.000
Max: 1.000
Min: -1.000
Mesh
Var: meshBLA
0.80
0.75
0.70
0.65
0.60
0.55
0.50
Y Axis
0.55
0.60
0.65
0.70
0.75
0.80
0.85
X Axis

Motivação



https://www.youtube.com/watch?v=DkrP-IrPydM

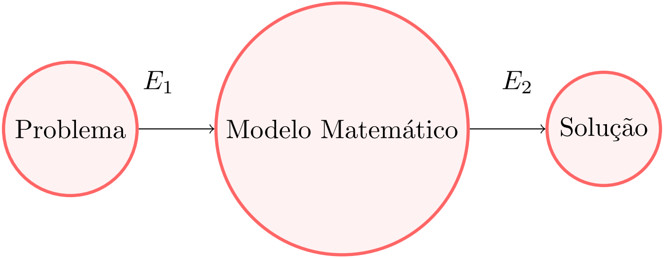
Motivação



https://www.youtube.com/watch?v=CKUIBw1SKHI

Motivação

Esquema do processo de solução:



E_1 - Modelagem (Formulação matemática) do problema: O objetivo é obter um conjunto de equações e relações que represente da maneira mais conveniente o problema de interesse.

E_2 - Etapa de solução, aqui obtida por meio de métodos numéricos.

A validação do modelo pode nos levar a uma terceira etapa, caso a solução não seja satisfatória.

Fundamentos

A taxa de variação de uma função $f(t)$, $f:[0,T]\to\mathbb{R}$, é dada pela razão

$$\frac{f(t_1)-f(t_0)}{t_1-t_0},$$

com $t_0,t_1\in[0,T]$. Fazendo $\Delta t=t_1-t_0$ e aplicando limite com $\Delta t\to 0$, temos a definição de derivada em t_0,

$$\frac{df(t_0)}{dt}=\lim_{\Delta t\to 0}\frac{f(t_0+\Delta t)-f(t_0)}{\Delta t}. \tag{1}$$

Dizemos que a função f é derivável em $[0,T]$ se o limite (1) existe para todo $t\in[0,T]$. A derivada de f também denotada por $f'(t)$.
Vamos estudar um método para aproximar derivada em um ponto t_0.

Fundamentos

- Um **algoritmo** é um processo sistemático para a resolução de um problema.
- O processo de desenvolvimento de algoritmos é importante para problemas a serem resolvidos em um computador.
- Um algoritmo computa uma *saída* (a resposta de um problema), a partir de uma *entrada* (as informações inicialmente conhecidas), que permitem determinar a solução do problema.
- Durante o processo de computação o algoritmo manipula dados gerados a partir de sua entrada, por meio de instruções.
- Um algoritmo implementado em uma linguagem de programação é o que chamamos de programa.
- Um **programa** é uma sequência de instruções que especificam como executar uma computação.

Fundamentos

A estrutura básica de um programa em diferentes linguagens de programação consiste em:

- **Entrada:** os dados de entrada são inicializados, recebidos por meio dos dispositivos de entrada ou de um arquivo.
- **Saída:** A reposta fornecida pelo algoritmo é transmitida para os dispositivos de saída ou salva em um arquivo.
- **Lógica e aritmética:** operações aritméticas e lógicas que resolvem o problema de interesse.
- **Condicional:** verifica se uma condição é satisfeita antes de executar uma sequência de instruções.
- **Repetição:** executa repetidas vezes um bloco de instruções, até que uma condição deixe de ser satisfeita.

Fundamentos

- Variáveis simples e estruturas de dados, como vetores, armazenam dados. Os elementos de vetores são identificados por índices.
- A atribuição de valores é indicada pelo símbolo $\leftarrow$.
- As seguintes declarações são empregadas para representar:
  - Condicional:
    se ... **então**
    se ... **então** ... **caso contrário**
  - Repetição:
    enquanto ... **faça**
    para ... **faça**

Fundamentos

Queremos verificar se uma pessoa pode doar sangue. Para isso ela precisa ter no mínimo 18 anos e pesar no mínimo 50 kg.

Algoritmo 1: Teste 1 para doação de sangue

Receba idade e peso
se *idade* ≥ 18 *e peso* ≥ 50 **então**
    Devolva "Possível Doadora";
senão
    Devolva "Não satisfaz as condições"

Fundamentos

O Algoritmo 2 recebe dois números inteiros a e b, com $b \neq 0$ e devolve o quociente da divisão a/b.

Algoritmo 2: Divisão inteira

Receba a e b
$c \leftarrow 0$
enquanto $a \geq b$ **faça**
    $c \leftarrow c + 1$
    $a \leftarrow a - b$
Devolva c

Vamos fazer a simulação do algoritmo com a entrada $a = 13$ e $b = 4$.

Exercício: Escreva um algoritmo que dado um número inteiro ($n \geq 0$) devolva a soma de seus algarismos.
Exemplo: Para $n = 123$ seu algoritmo deve devolver 6.
Dica: utilize o operador % resto da divisão.

Fundamentos

Mais um exemplo ...
O Algoritmo 3 recebe uma sequência e inverte a ordem dos termos dela.

Algoritmo 3: Inversão de uma sequência

Receba S um vetor de tamanho $n+1$
para $i = 0, \dots, \lfloor n/2 \rfloor$ **faça**
    temp $\leftarrow S[i]$
    $S[i] \leftarrow S[n-i]$
    $S[n-i] \leftarrow$ temp

Vamos fazer a simulação do algoritmo com a entrada $S = [-1, 2, 0, 3, 7]$.

Exercício: Escreva um algoritmo que dado um número inteiro $(n \geq 0)$ verifique se n é primo.
Exemplo: Para $n = 4$ seu algoritmo deve devolver 0.
Para $n = 7$ seu algoritmo deve devolver 1.

Modelo Matemático

Um primeiro modelo de **crescimento populacional** considera que a taxa de crescimento de uma população é proporcional a população total,

$$\frac{P(t_0 + \Delta t) - P(t_0)}{\Delta t} = kP(t_0), \tag{2}$$

sendo k uma constante de proporcionalidade e $P(t_0) = P_0$ a população inicial. Aplicando limite quando $\Delta t \to 0$, obtemos um problema de valor inicial,

$$\frac{dP}{dt} = kP(t), \quad t \in [t_0, t_n], P(t_0) = P_0. \tag{3}$$

A solução desse problema é $P(t) = P_0 e^{kt}$, crescimento exponencial.

Aplicação:
Uma espécie de bactéria, e condições ideais, cresce a uma taxa de 30% a cada 1 hora. Supondo que a população inicial é de 100, qual a quantidade de bactérias após 2 horas?

Modelo Matemático

Um segundo modelo de **crescimento populacional** considera que a população cresce até um limite, tende a estabilizar. Assim a taxa de crescimento é dada por,

$$\frac{dP}{dt} = k\left(1 - \frac{P(t)}{M}\right)P(t), \quad t \in [t_0, t_n], P(t_0) = P_0, \tag{4}$$

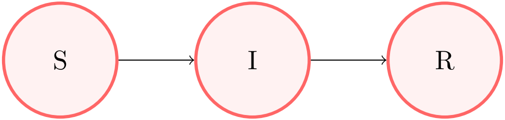
sendo k uma constante positiva e M a capacidade máxima populacional. Inicialmente $P(t) << M$, note que a medida que $P(t)$ se aproxima de M, a taxa de crescimento diminui.

Aplicação:
Considere o exemplo anterior com uma capacidade populacional $M = 950$.

Modelo Matemático

- Considere uma vila em que uma ou mais pessoas estão com uma infecção respiratória, como a gripe. A doença pode se espalhar rapidamente ou terminar logo. Nosso interesse é estimar, e de certa forma controlar, quantas pessoas serão acometidos pela doença.
- Vamos dividir as pessoas em grupos, S (Suscetíveis), I (Infectados) e R (Removidos).

Dinâmica de espalhamento da doença.



- O parâmetro β é a taxa de infecção.
- O parâmetro γ é a taxa de recuperação.

Modelo Matemático

Estimativa do parâmetro β:

- Durante um intervalo de tempo T foram contabilizados m encontros entre pares S e I, dentre os n possíveis pares de pessoas dos dois grupos. A probabilidade por unidade de tempo é $\mu = m/(nT)$.
- O valor esperado de encontros S e I será μSI. Apenas uma fração destes encontros resultará em uma pessoa infectada $p/e\mu$, sendo p o número de infectados a cada e encontros. Assim, $\beta = (p/e)\mu$.

Estimativa do parâmetro γ:

- Durante um intervalo de tempo T, m pessoas de cada n infectados se recuperam.
- Assim, $\gamma = m/(nT)$ é a probabilidade que um indivíduo se recupere em um intervalo de tempo.

Modelo Matemático

- A taxa de diminuição de pessoas em S, no intervalo Δt é dada por,

$$\frac{S(t_k + \Delta t) - S(t_k)}{\Delta t} = -\beta S(t_k) I(t_k). \tag{5}$$

- A taxa de variação de pessoas no grupo I é dada por,

$$\frac{I(t_k + \Delta t) - I(t_k)}{\Delta t} = \beta S(t_k) I(t_k) - \gamma I(t_k). \tag{6}$$

- A taxa de variação de pessoas no grupo R é dada por, a variação de indivíduos no grupo I é dada por,

$$\frac{R(t_k + \Delta t) - R(t_k)}{\Delta t} = \gamma I(t_k), \tag{7}$$

com $t_k \in [t_0, t_n]$ e as condições iniciais $S(t_0) = S_0$, $I(t_0) = I_0$ e $R(t_0) = R_0$.

Modelo Matemático

Quando fazemos, $\Delta t \to 0$ em (5), (6) e (7), obtemos as taxas de variação instantânea, dadas por

$$
\begin{aligned}
\frac{dS}{dt} &= -\beta S I, \\
\frac{dI}{dt} &= \beta S I - \gamma I, \\
\frac{dR}{dt} &= \gamma I,
\end{aligned}
\tag{8}
$$

sendo S, I e R funções em $t \in [t_0, t_n]$, as incógnitas do problema e S_0, I_0 e R_0 as condições iniciais do problema.

Método Numérico

Dado um problema de valor inicial,

$$\frac{dy}{dt} = f(t, y),$$

com $f(t, y)$ e $y(t_0)$ dados. Queremos determinar $y(t)$, com $t \in [t_0, t_n]$

- Primeiro passo é discretizar o intervalo $[t_0, t_n]$, ou seja, dividir o intervalo em n subintervalos igualmente espaçados (de tamanho Δt), com $t_0 < t_1 < \ldots < t_n$.
- Para cada t_k vamos obter uma aproximação y_k de $y(t_k)$.
- O Método de Euler é definido por,

$$\begin{cases} y_0 & = y(t_0), \\ y_{k+1} & = y_k + \Delta t f(t_k, y_k), \end{cases} \tag{9}$$

com $k = 0, 1, 2, \ldots$

Método Numérico

Algoritmo 4: Método de Euler

Dados y_0, t_0, t_n e $f(t, y)$
Receba n (número de subintervalos)
$\Delta t \leftarrow (t_n - t_0)/n$
$t[0] \leftarrow t_0$
para $k = 0, \ldots, n-1$ **faça**
  $t[k+1] \leftarrow t[k] + \Delta t$
  $y[k+1] \leftarrow y[k] + \Delta t f(t[k], y[k])$

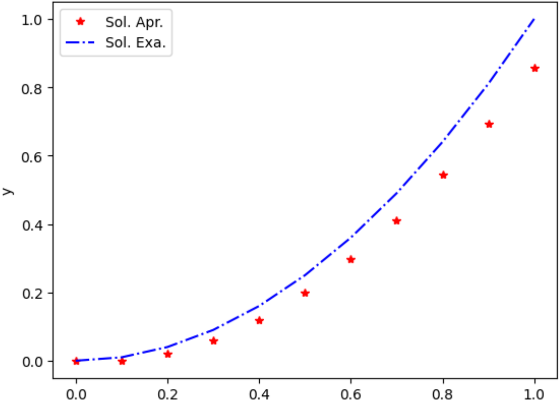
- A **ordem** do erro global E do Método de Euler é $E = \mathrm{O}(\Delta t)$. Ou seja, $E \leq C\Delta t$.
- Vamos verificar se o método está implementado corretamente, resolvendo um problema com solução conhecida.

Método Numérico

Considere o problema,

$$\begin{cases} y(0) & = 0, t \in [0,1] \\ \dfrac{dy}{dt} & = y^2(t) + 2y(t) - t^4, \end{cases} \tag{10}$$

cuja solução exata é $y(t) = t^2$. Vamos simular o Algoritmo 4 com $\Delta t = 0.1$.



Método Numérico

A tabela apresenta o erro em $t = 1$ para diferentes valores de n.

n	Erro	Razão
8	0.17472813578147073	—
16	0.09473515363834206	1.844
32	0.04966799130253441	1.907
64	0.02548427415316057	1.949
128	0.012915648378055478	1.973

Verificação Numérica da ordem: $E(n) = \mathrm{O}(\Delta t^p)$ e $E(2n) = \mathrm{O}((\Delta t/2)^p)$

$$\frac{E(n)}{E(n/2)} \approx \frac{\Delta t^p}{\frac{\Delta t^p}{2^p}} = 2^p$$

Exemplo

`https://github.com/pccalegari/EHH/blob/main/exemplo_euler.ipynb`

Experimento Numérico

Crescimento populacional: Problema 1

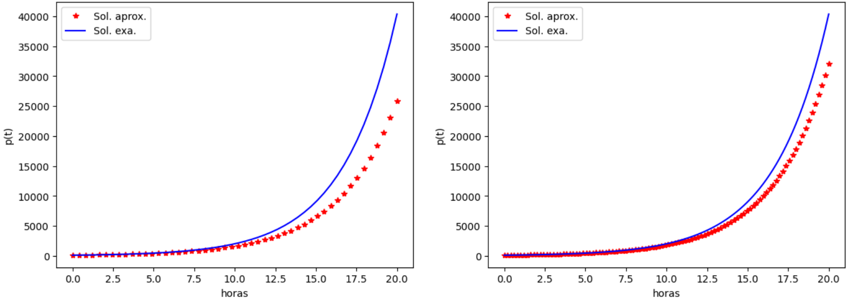
$$\frac{dP}{dt} = kP(t), \quad t \in [t_0, t_n], P(t_0) = P_0. \tag{11}$$

Dados $k = 0.3$, $P_0 = 100$ e $t_n = 50$. O método de Euler aplicado a esse problema é dado por,

$$\begin{cases} P_0 & = 100, \\ P_{i+1} & = P_i + \Delta t(kP_i), \quad i = 0, 1, \dots, n-1. \end{cases} \tag{12}$$

Experimento Numérico

Intervalo $[t_0, t_n]$ dividido em 50 e 100 subintervalos, respectivamente.



Experimento Numérico

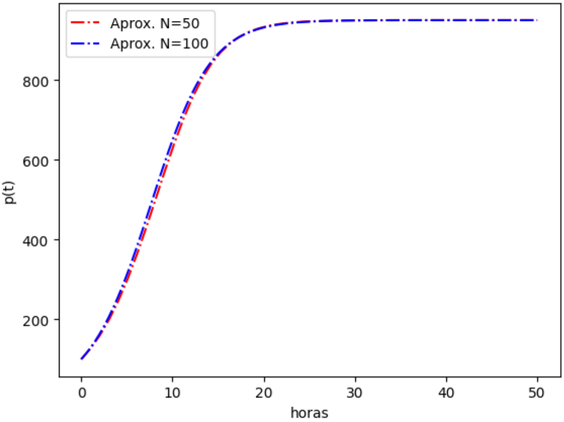
Crescimento populacional: Problema 2

$$\frac{dP}{dt} = k\left(1 - \frac{P(t)}{M}\right)P(t), \quad t \in [t_0, t_n], P(t_0) = P_0. \tag{13}$$

Dados $k = 0.3$, $P_0 = 100$, $M = 950$ e $t_n = 20$. O método de Euler aplicado a esse problema é dado por,

$$\begin{cases} P_0 & = 100, \\ P_{i+1} & = P_i + \Delta t\left(1 - \dfrac{P_i}{M}\right)kP_i, \quad i = 0, 1, \ldots, n-1. \end{cases} \tag{14}$$

Experimento Numérico



Experimento Numérico

Espalhamento de uma doença: Modelo SIR

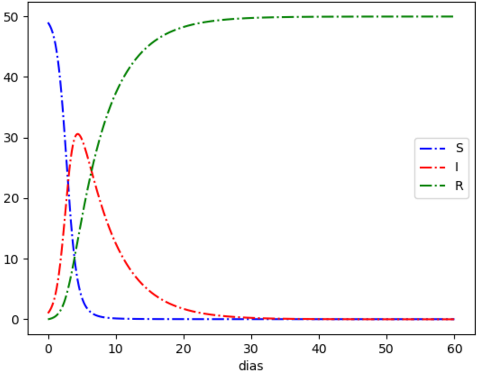
$$\begin{aligned}
\frac{dS}{dt} &= -\beta S I,\ S_0 = S(t_0),\\
\frac{dI}{dt} &= \beta S I - \gamma I,\ I_0 = I(t_0),\\
\frac{dR}{dt} &= \gamma I,\ R_0 = R(t_0).
\end{aligned} \tag{15}$$

Aplicando o Método de Euler em (15) obtemos,

$$\begin{aligned}
S_{k+1} &= S_k + \Delta t(-\beta S_k I_k),\\
I_{k+1} &= I_k + \Delta t(\beta S_k I_k - \gamma I_k),\\
R_{k+1} &= R_k + \Delta t \gamma I_k.
\end{aligned} \tag{16}$$

Experimento Numérico

Condições iniciais $S_0 = 49$, $I_0 = 1$ e $R_0 = 0$, $t_n = 60$ dias. Os parâmetros $\beta = 10/(40 \cdot 8 \cdot 24)$ e $\gamma = 3/(15 \cdot 24)$. Ou seja, em 24 horas de 40 suscetíveis e 8 infectados passou-se a 30 e 18, respectivamente. Além disso, de 15 indivíduos infectados, 3 se recuperaram durante um dia.



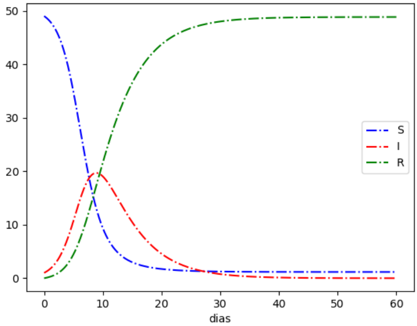
O pico de aproximadamente 30 infectados ocorre no dia 4.

Experimento Numérico

Incluindo um fator de isolamento de α, separamos cada grupo em dois: $\dot{S} = \alpha S_k$ e $\ddot{S} = (1-\alpha)S_k$, pessoas que atendem e que não atendem o isolamento, respectivamente. O que muda no modelo,

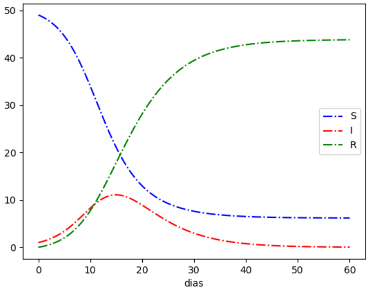
$$\begin{aligned} S_{k+1} &= S_k + \Delta t(-\beta \ddot{S}\ddot{I}), \\ I_{k+1} &= I_k + \Delta t(\beta \ddot{S}\ddot{I} - \gamma I_k), \\ R_{k+1} &= R_k + \Delta t \gamma I_k. \end{aligned} \tag{17}$$

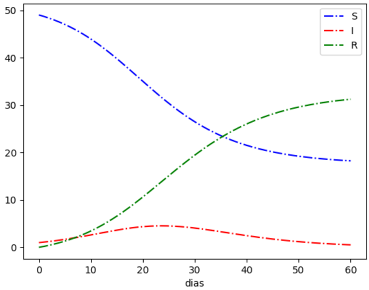
Com $\alpha = 30\%$, o pico de aproximadamente 20 infectados ocorre no dia 8.



Experimento Numérico

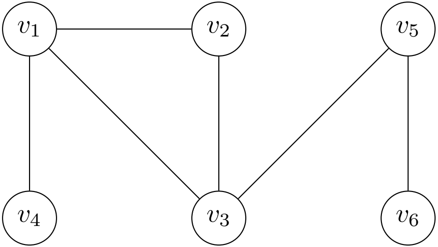
O pico de aproximadamente 11 infectados (com 45% de isolamento) ocorre no dia 14. Já o pico de aproximadamente 5 infectados (com 55%) ocorre no dia 23.





Grafos

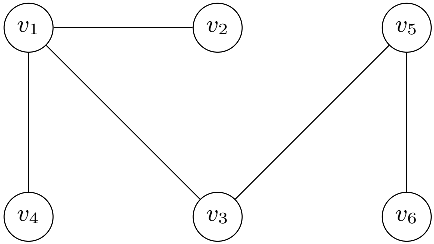
- Um **Grafo** é definido como o par de conjuntos $G = (V, A)$.
- O conjunto de vártices $V = \{v_1, v_2, \dots, v_n\}$.
- O conjunto de arestas $A \subset \{v_iv_j, \text{ tal que } v_i, v_j \in V\}$. Cada aresta é definida como um par de vértices.



$V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ e $A = \{v_1v_2, v_1v_3, v_1v_4, v_2v_3, v_3v_5, v_5v_6\}$.

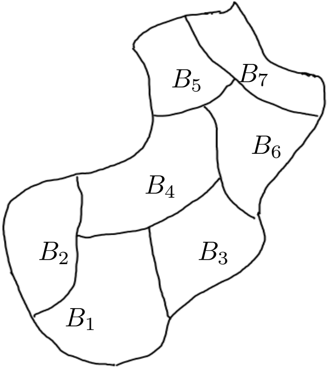
Grafos

- Um **caminho** entre dois vértices v_i e v_j é uma sequência de vértices e arestas que conecta v_i a v_j.
- Um **grafo é conexo** quando existe um caminho entre quaisquer dois vértices do grafo.
- Um **ciclo** em um grafo é um caminho fechado que sai de v_i e retorna a v_i.
- Um grafo é uma **árvore** se ele é conexo e não contém ciclos.
- Uma **árvore geradora** de um grafo conexo G é um subgrafo de G que contém todos os vértices de G e é uma árvore.



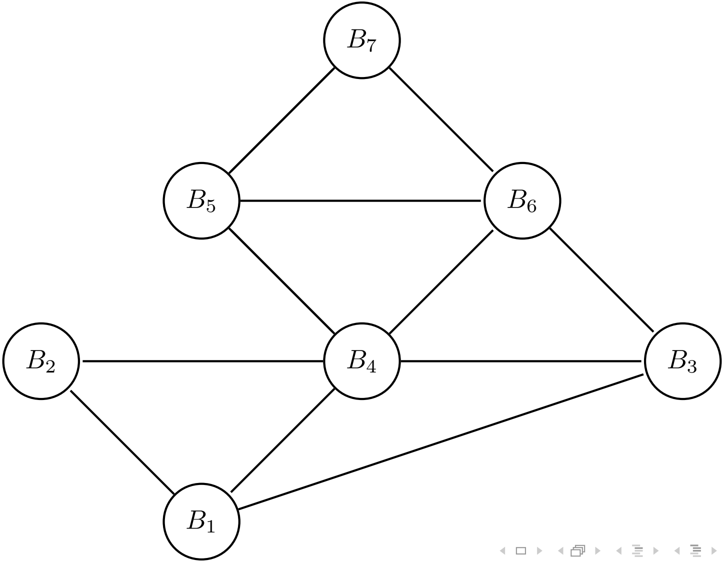
Grafos

Vamos representar um mapa por meio de um grafo.



Grafos

Cada bairro será representado por um vértice e bairros vizinhos são conectados por arestas.



Grafos

- Os grupos são divididos $\dot{S}$ (atendem o isolamento) e $\ddot{S}$ (não atendem o distanciamento).
- Aos grupos $\dot{S}$, $\dot{I}$ e $\dot{R}$ incluimos um fator de movimentação λ.
- A interação entre os grupos só ocorre entre bairros vizinhos.
- Para cada vértice i, temos um sistema discreto

$$\begin{aligned} S_{k+1}^i &= S_k^i + \Delta t(-\beta)X_k^i, \\ I_{k+1}^i &= I_k^i + \Delta t(\beta X_k^i - \gamma I_k^i), \\ R_{k+1}^i &= R_k^i + \Delta t(\gamma I_k^i), \end{aligned} \tag{18}$$

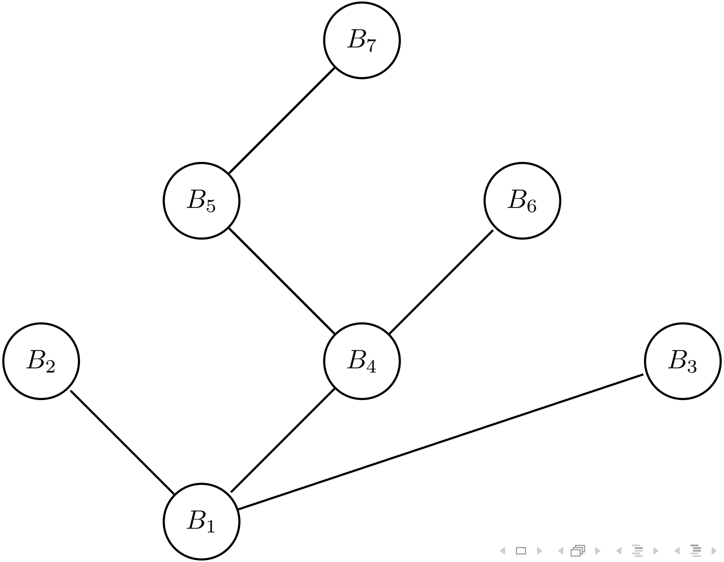
sendo βX_k^i contabiliza os infectados do vértice i dentre os possíveis encontros entre infectados e suscetíveis que circulam no vértice i.
O total de pessoas de cada grupo em t_k é dado por,

$$S_k = \sum_{i=1}^{N_v} S_k^i, \quad I_k = \sum_{i=1}^{N_v} I_k^i, \quad R_k = \sum_{i=1}^{N_v} R_k^i, \tag{19}$$

sendo N_v o número de vértices do grafo.

Grafos

Estratégias para conter o espalhamento: Remoção de arestas do grafo de infecção, sem deixá-lo desconexo, realizada manualmente.



Grafos

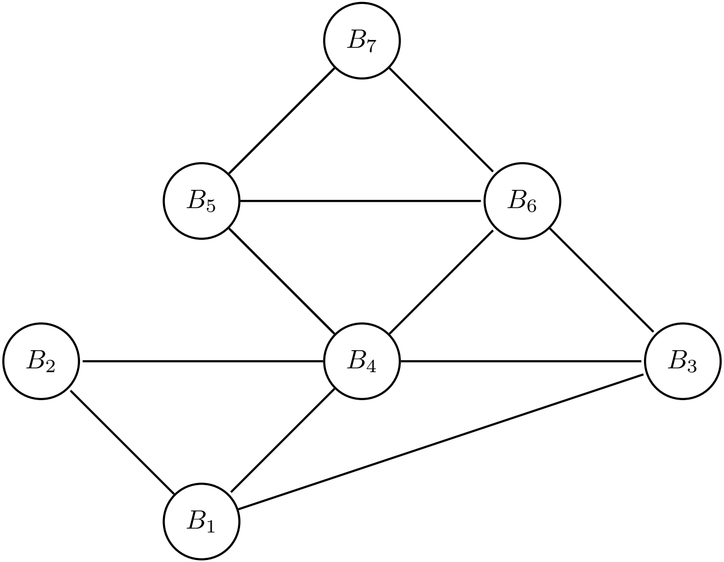
- Os algoritmos de busca em grafos podem ser utilizados para obter árvores geradoras. Tais algoritmos visitam vértices em uma ordem definida. A partir disso, podemos criar a árvore ao conectar vértices visitados ao vértice visitado na etapa anterior.
- Na Busca em Profundidade visitamos todos os vértices de um caminho até chegar ao final. A partir daí, voltamos por esse caminho, entrando em todos os outros caminhos que encontrarmos nessa volta. Ao entrar em um caminho, seguimos até não encontrarmos a saída.

Algoritmo 5: Busca em Profundidade

```
Explorar(Grafo G = (V, E), v ∈ V):
    visitado(v) ← true
    para cada vizinho u de v faça
        se não visitado(u) então
            Explorar(u)
```

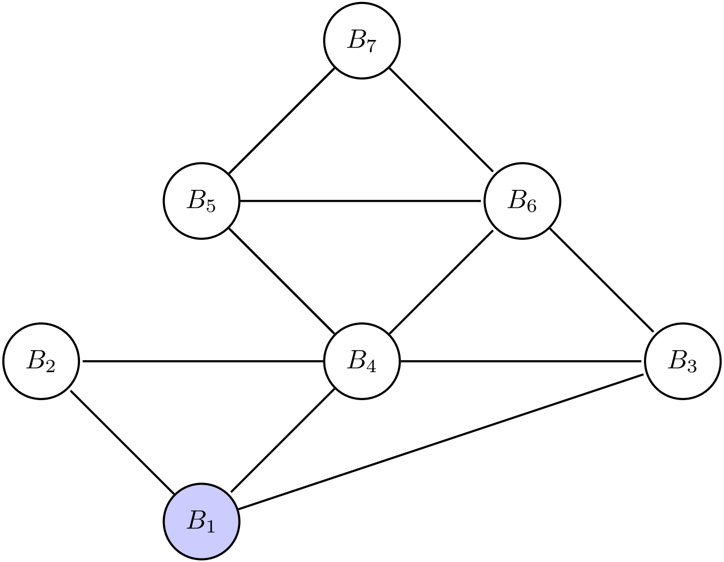
Grafos

Busca em Profundidade no grafo associado ao mapa de bairros.



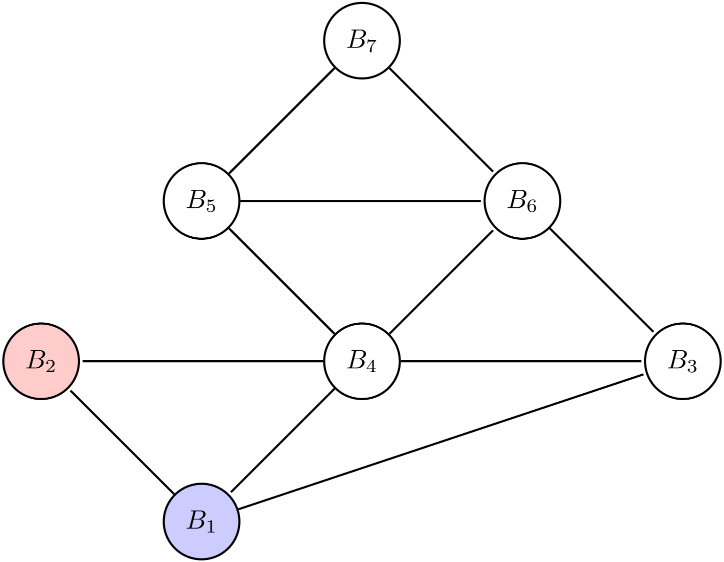
Grafos

Busca em Profundidade no grafo associado ao mapa de bairros.



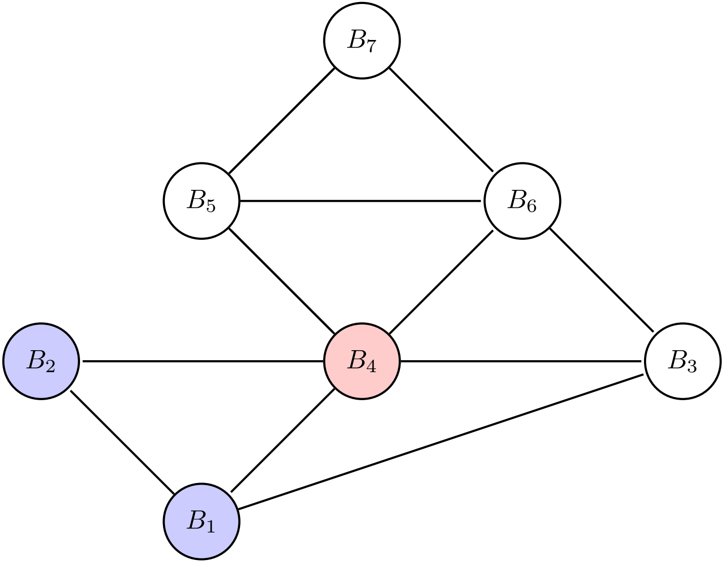
Grafos

Busca em Profundidade no grafo associado ao mapa de bairros.



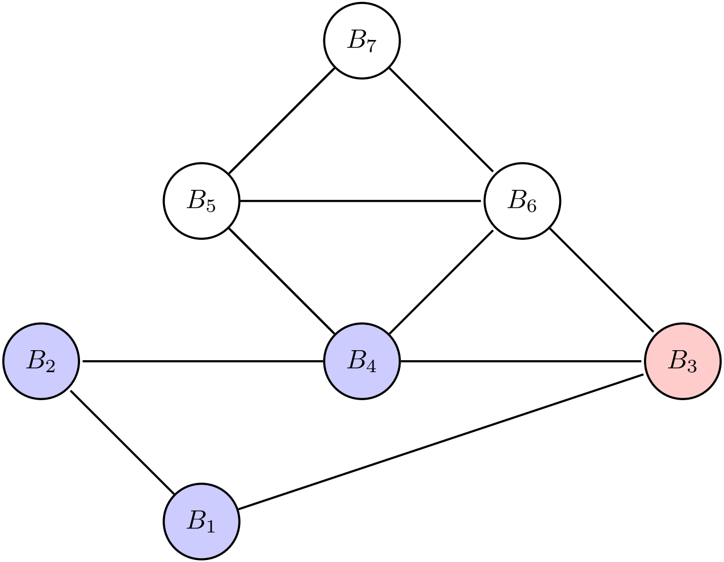
Grafos

Busca em Profundidade no grafo associado ao mapa de bairros.



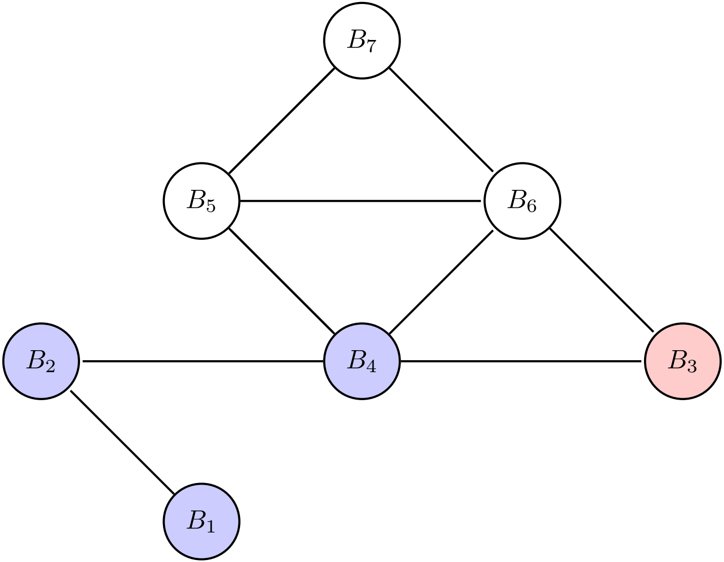
Grafos

Busca em Profundidade no grafo associado ao mapa de bairros.



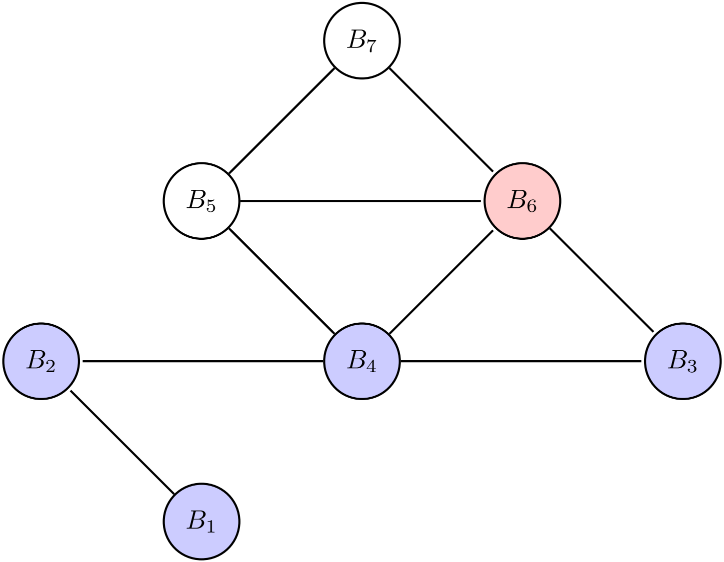
Grafos

Busca em Profundidade no grafo associado ao mapa de bairros.



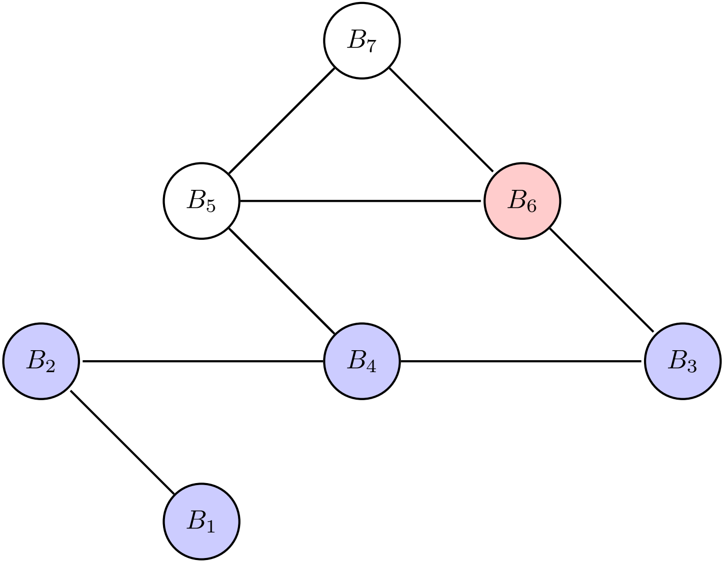
Grafos

Busca em Profundidade no grafo associado ao mapa de bairros.



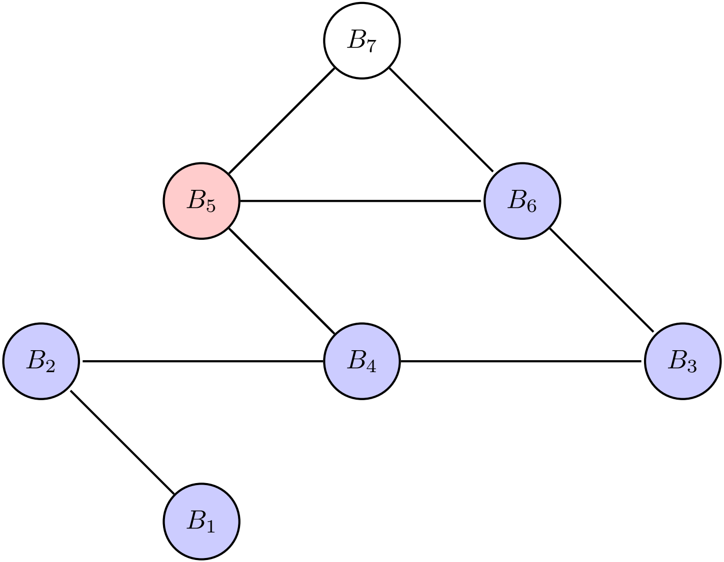
Grafos

Busca em Profundidade no grafo associado ao mapa de bairros.



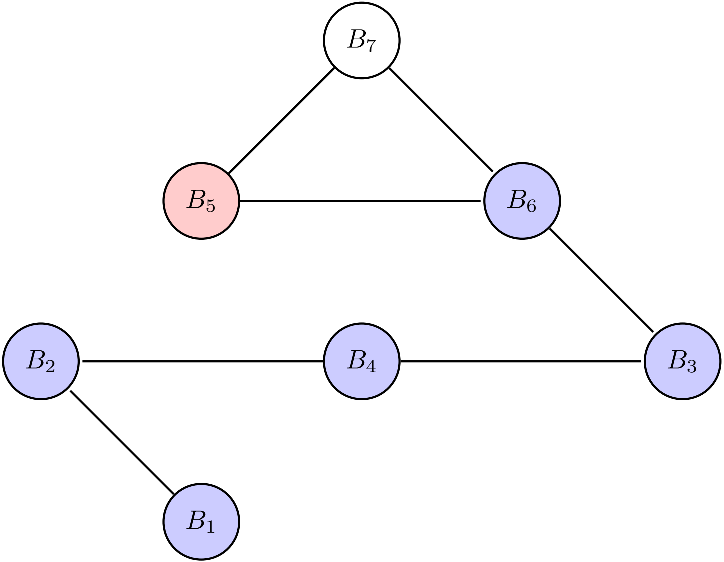
Grafos

Busca em Profundidade no grafo associado ao mapa de bairros.



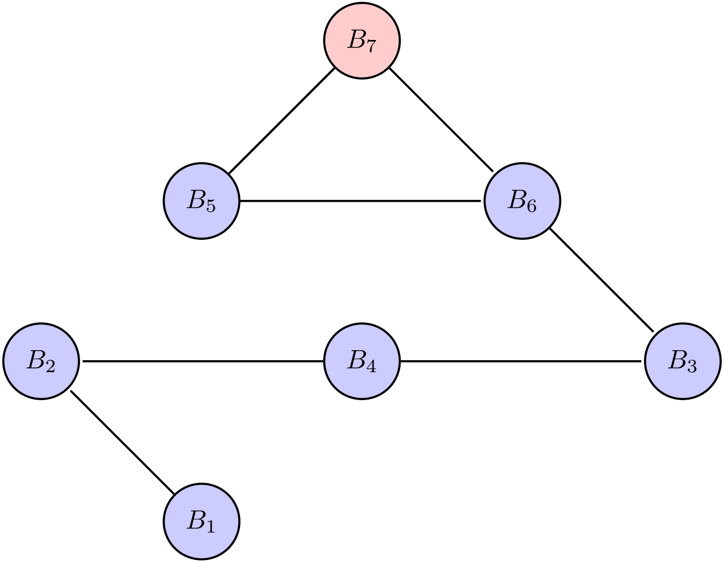
Grafos

Busca em Profundidade no grafo associado ao mapa de bairros.



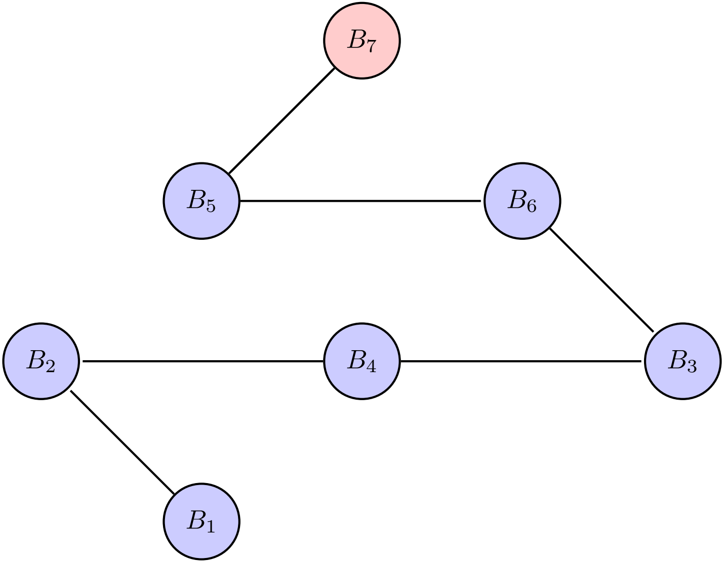
Grafos

Busca em Profundidade no grafo associado ao mapa de bairros.



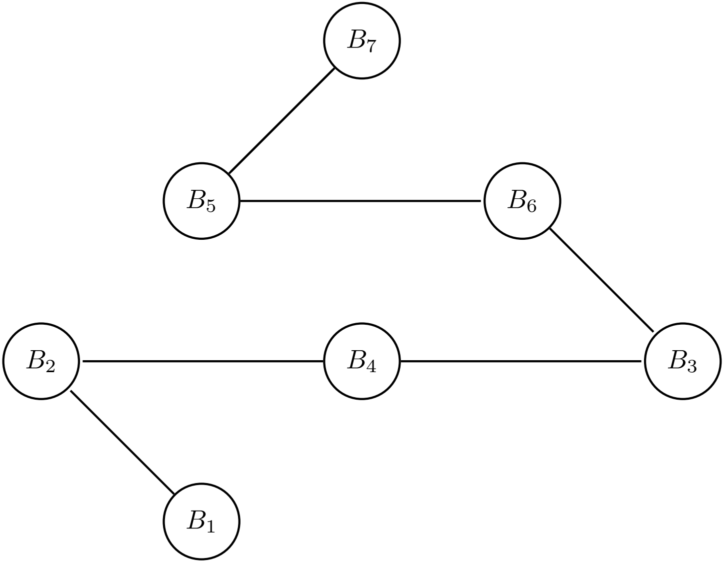
Grafos

Busca em Profundidade no grafo associado ao mapa de bairros.



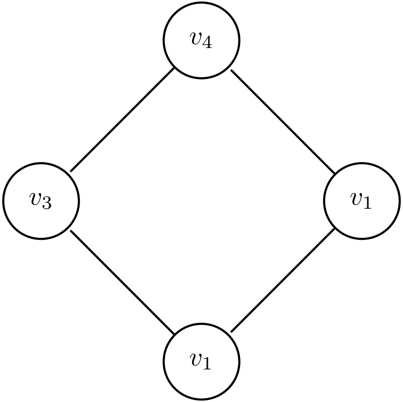
Grafos

Busca em Profundidade no grafo associado ao mapa de bairros.



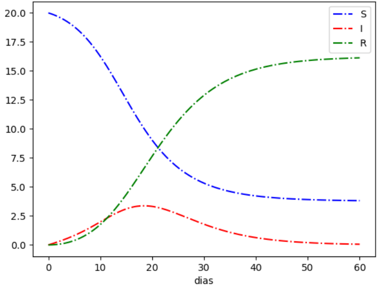
Experimentos Numéricos

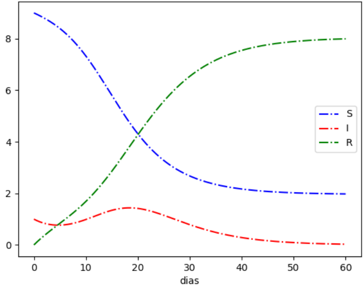
Experimento 1: Vila representada por um grafo com 4 vértices e sem isolamento.
v_1: $S_0 = 20, I_0 = 0$ e $R_0 = 0$. v_2: $S_0 = 10, I_0 = 1$ e $R_0 = 0$.
v_3: $S_0 = 10, I_0 = 0$ e $R_0 = 0$. v_4: $S_0 = 10, I_0 = 0$ e $R_0 = 0$.



Experimentos Numéricos

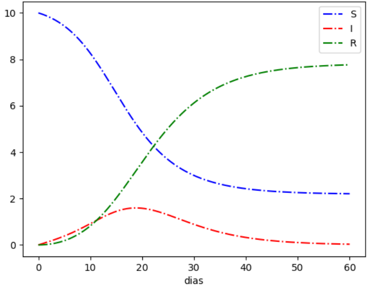
Evolução dos vértices 1 e 2:

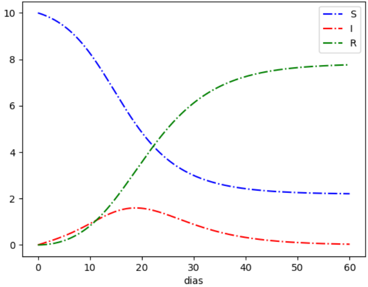




Experimentos Numéricos

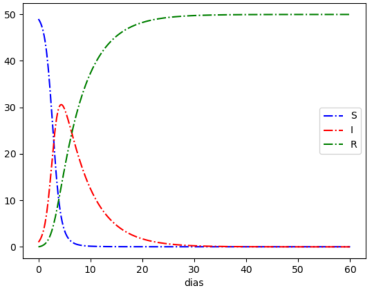
Evolução dos vértices 3 e 4:

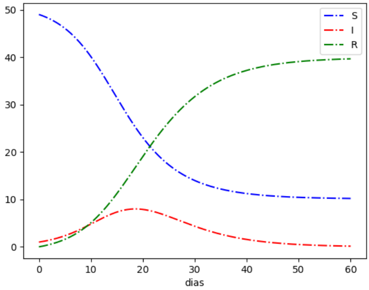




Experimentos Numéricos

Comparação entre o experimento com os valores totais.



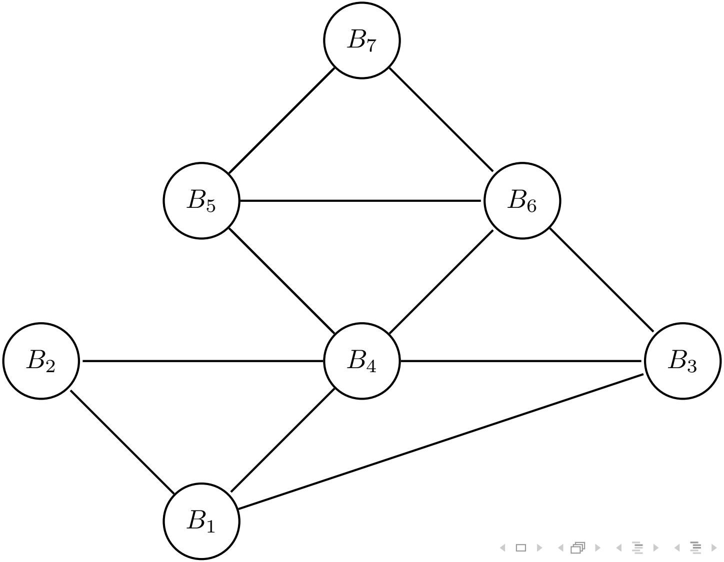


Pico de infectados: 30 no dia 4 e 8 no dia 18.

Experimentos Numéricos

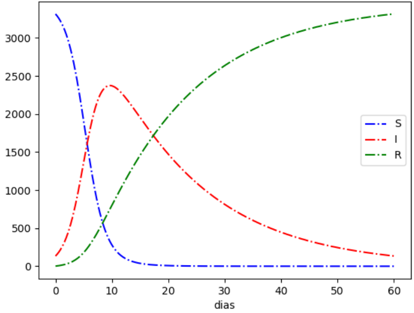
Experimento 2:

- O mapa com 7 bairros com $\alpha = 30\%$.



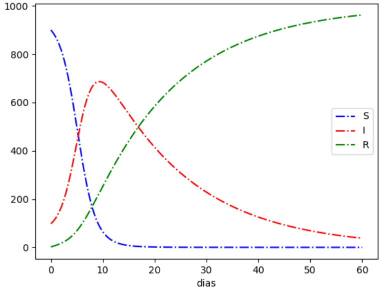
Experimentos Numéricos

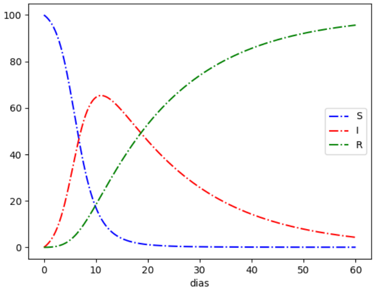
- Os parâmetros β contabilizando a probabilidade de 0.2 encontros e 1 infectado a cada 100 encontros e γ contabilizando 6 recuperados a cada 100 infectados por dia.
- Os fatores de movimentação: $\lambda_S = 0.4$, $\lambda_I = 0.1$ e $\lambda_R = 0.6$.
- $S_0 = 3315$, $I_0 = 134$ e $R_0 = 2$ distribuidos pelos 7 vértices.
- O pico de aproximadamente 2400 infectados ocorreu no dia 9.



Experimentos Numéricos

- Vértice 4: $S_0 = 900$, $I_0 = 98$ e $R_0 = 2$. Com $I_p \approx 686$ no dia 9.
- Vértice 7: $S_0 = 100$, $I_0 = 0$ e $R_0 = 0$. Com $I_p \approx 65$ no dia 11.

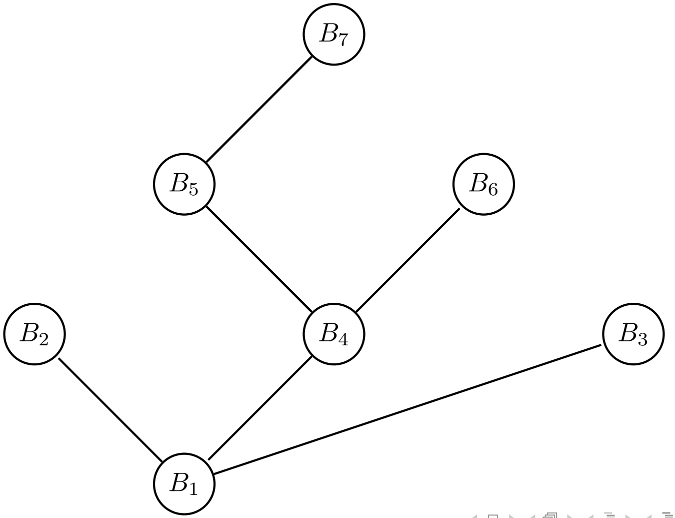




Experimentos Numéricos

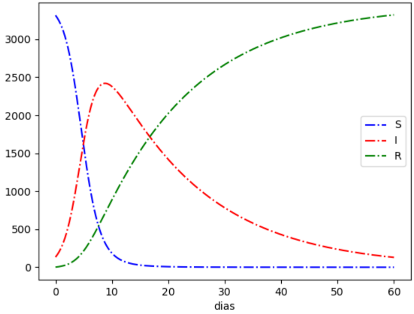
Experimento 3:

- Árvore geradora (manual) do grafo associado ao mapa com 7 bairros e mesmos dados do experimento 2.



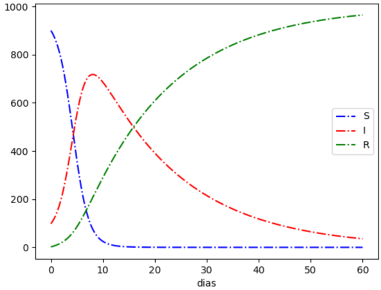
Experimentos Numéricos

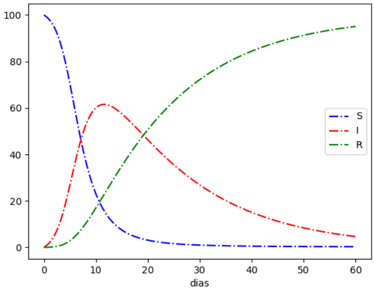
- O pico de aproximadamente 2420 infectados ocorreu no dia 8.



Experimentos Numéricos

- Vértice 4: $S_0 = 900$, $I_0 = 98$ e $R_0 = 2$. Com $I_p \approx 717$ no dia 8.
- Vértice 7: $S_0 = 100$, $I_0 = 0$ e $R_0 = 0$. Com $I_p \approx 61$ no dia 11.

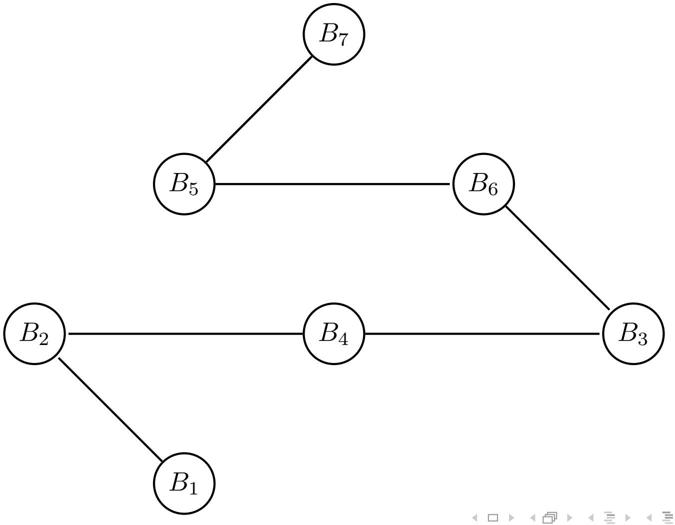




Experimentos Numéricos

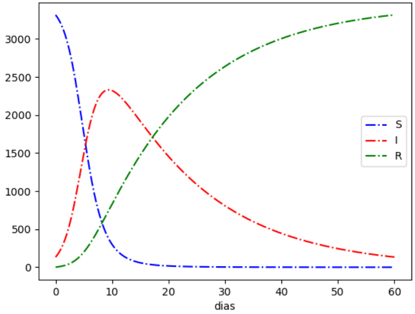
Experimento 4:

- Árvore geradora (busca em profundidade) do grafo associado ao mapa com 7 bairros e mesmos dados do experimento 2.



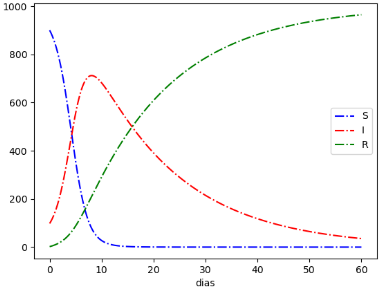
Experimentos Numéricos

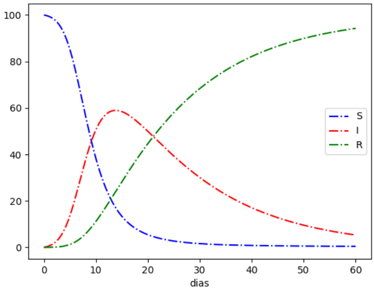
- O pico de aproximadamente 2330 infectados ocorreu no dia 9.



Experimentos Numéricos

- Vértice 4: $S_0 = 900$, $I_0 = 98$ e $R_0 = 2$. Com $I_p \approx 711$ no dia 8.
- Vértice 7: $S_0 = 100$, $I_0 = 0$ e $R_0 = 0$. Com $I_p \approx 58$ no dia 13.





Referências

[1] Lima, E. L. Análise real, volume 1 ($6^{\underline{a}}$ edição). Coleção Matemática Universitária, 2002.

[2] Humes, A. F. P *et al.* Noções de cálculo numérico. McGrawHillP, 1984.

[3] Os modelos de `https://www.fc.unesp.br/Home/Departamentos/Matematica/revistacqd2228/v02n02a09-os-modelos-de-crescimento-populacional.pdf`

[4] Linge, S. e Langtangen, H. P. Programming for computations - Python: A gentle introduction to numerical simulations with Python. Springer Nature, 2016.

[5] Feofiloff, P. Kohayakawa, Y. e Wakabayashi, Y. Uma introdução sucinta à teoria dos grafos, 2011.

Referências

[6] Caninas, P. G., Calegari, P. C. e Franco, Á. Um estudo sobre árvores geradoras como alternativas conexas para o controle de doença em um grafo de infecção. Simpósio Brasileiro de Pesquisa Operacional, 2023.

[7] Szwarcfiter, J. L. e Markezon. Estruturas de Dados e seus Algoritmos. LCT, 2010.

[8] Franco, A. J. P. Epidemic models with restricted circulation and social distancing on some network topologies. ACRI - Cellular Automata for Research and and Industry, 2021.

[9] Morimoto, C. H. e de Pinha Jr, J. C. Introdução à Computação com Python: um curso interativo. `https://panda.ime.usp.br/cc110/static/cc110/index.html`.

[10] Morimoto, C. H. e Hashimoto, R. F. Introdução à Cincia da Computação em C. `https://www.ime.usp.br/~hitoshi/introducao/`.