Um Modelo Epidemiológico Sobre Redes

Primeiros Resultados

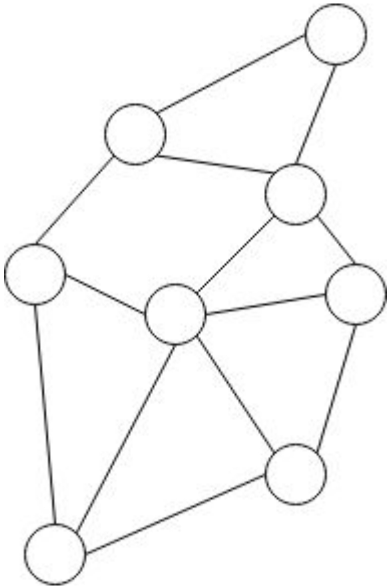
Motivação

- Prever a trajetória de uma infecção
- Importante para decidir medidas de prevenção!

- Nossa abordagem busca usar programação não-linear para otimizar uma topologia de um grafo representando uma cidade, de modo que o número de infectados seja o mínimo.

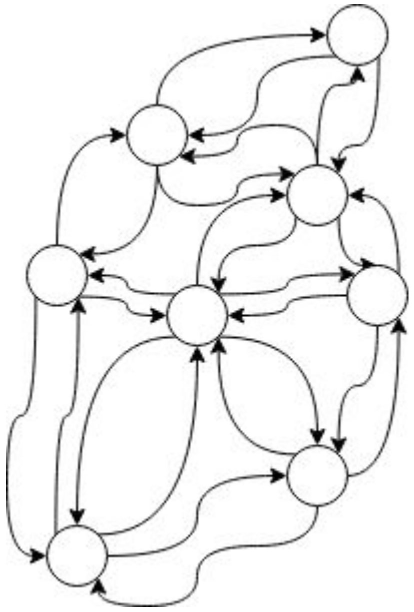
O Nosso Problema

- Grafo representando uma cidade
- Os vértices correspondem a bairros
- As arestas correspondem às ruas que ligam os bairros
- Cada bairros tem uma população de suscetíveis, infectados e recuperados



O Nosso Problema

- Pessoas de um bairro podem se mover para bairros adjacentes
- Para cada rua entre um bairro i e um bairro j, temos valores entre 0 e 1 que correspondem a porcentagem da população de i que se locomove para j e vice versa.



O Nosso Problema

- Pessoas circulando em um vértice
- Por exemplo, com as populações:

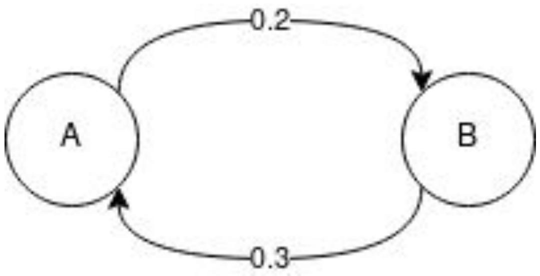
  A: S = 100; I = 10; R = 0;

  B: S = 200; I = 10; R = 0;

- Teríamos circulando em cada vértice na próxima iteração:

  A: S = 140; I = 11; R = 0;

  B: S = 160; I = 9; R = 0;



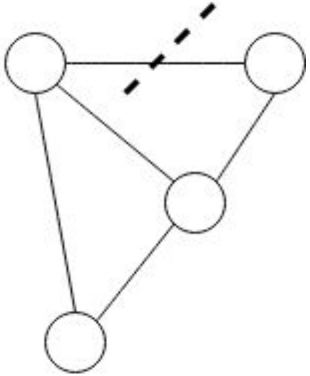
O Nosso Problema

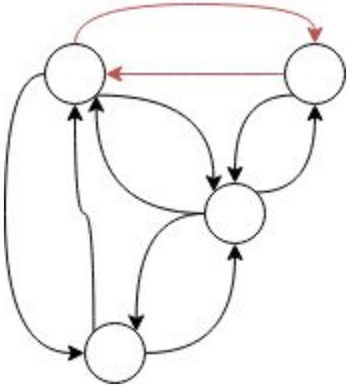
- Com as pessoas circulando em um vértice, temos um número esperado de encontros entre suscetíveis e infectados.
- Novas infecções de acordo com a **virulência**
- Recuperações de acordo com a **taxa de recuperação**

$$E = S_P * \frac{I}{S+I+R}$$

O Nosso Problema

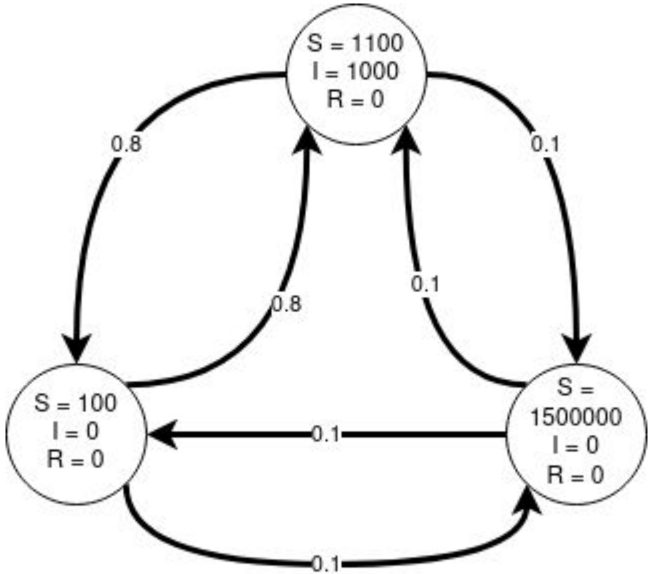
- Temos a opção de bloquear estradas
- Não há trânsito por uma via bloqueada
- Pessoas que atravessariam a estrada ficam no seu vértice
- Nosso objetivo: minimizar o número de infectados.
- **Mas queremos manter o grafo conexo!**





O Nosso Problema

- A árvore geradora mínima não é necessariamente a melhor opção, especialmente para grafos “desbalanceados”
- Nossa abordagem: programação não-linear



O Nosso Problema

- Nossas variáveis: bloquear cada aresta ou não
- Nós nos restringimos a uma iteração
- Solver SCIP através da interface JuMP, uma linguagem para definir modelos de otimização embutida em Julia

- A árvore geradora mínima pode ser um bom ponto de partida

- Mas como garantir a conexidade?

Garantir a conectividade

Usamos a ideia de **fluxo** para garantir a conectividade.

Mantemos um conjunto de arcos entre vértices que tem uma aresta entre eles no grafo original.

Fluxo s-arborescente

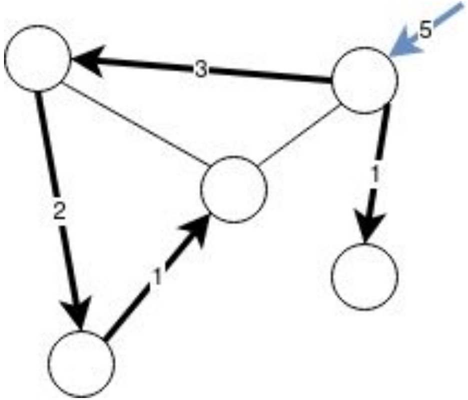
Duas condições:

- Para o vértice s (a fonte):

$$\sum_{(s,j)\in A} f_{(s,j)} = n - 1$$

- Para cada vértice i, que não seja a fonte:

$$\sum_{(i,j)\in A} f_{i\to j} + 1 = \sum_{(j,i)\in A} f_{j\to i}$$



Teorema

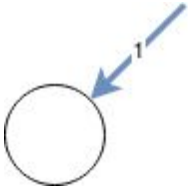
- Sejam G um grafo conexo e T uma árvore geradora de G. Para todo vértice v em T, existe um fluxo v-arborescente que gera a árvore T.

- A implicação é que podemos escolher qualquer vértice como fonte.

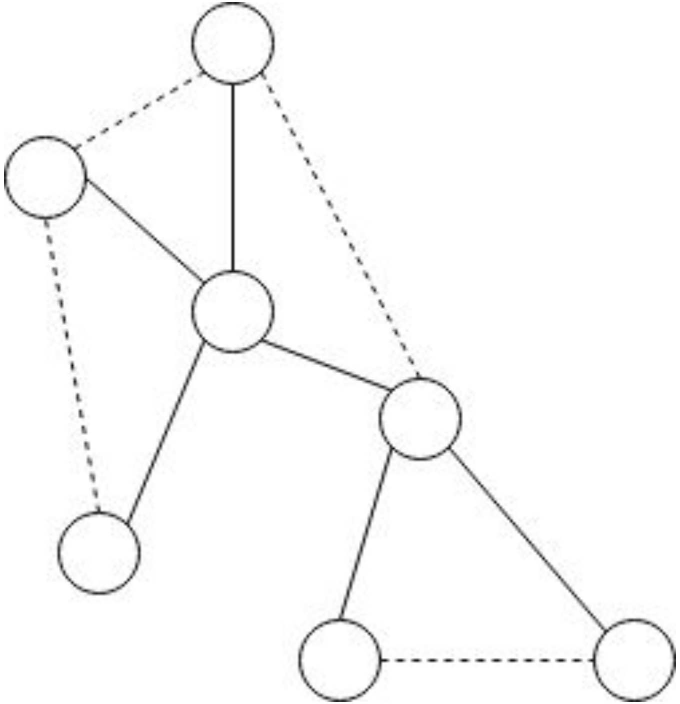
Prova do Teorema

- Caso Base

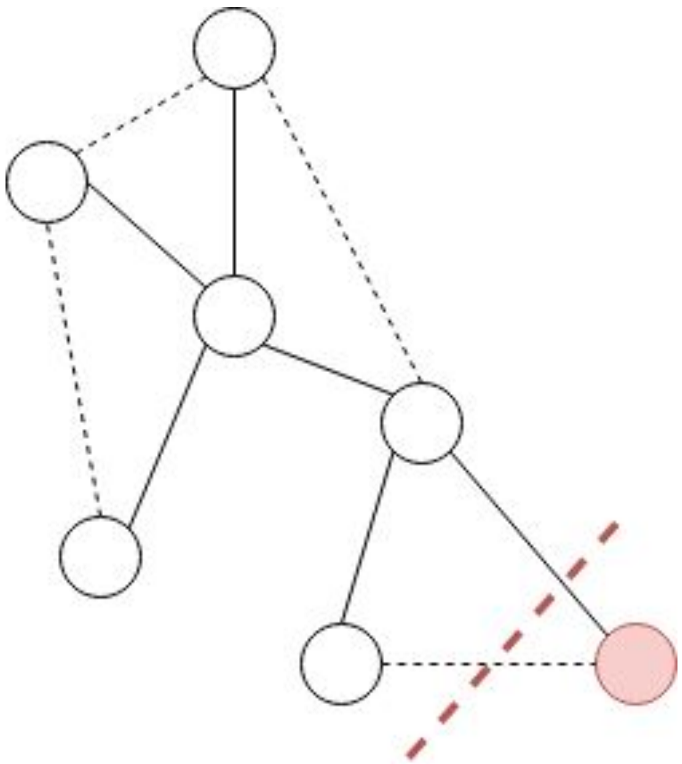
  Note que tomamos fluxos que não existem como valendo 0.



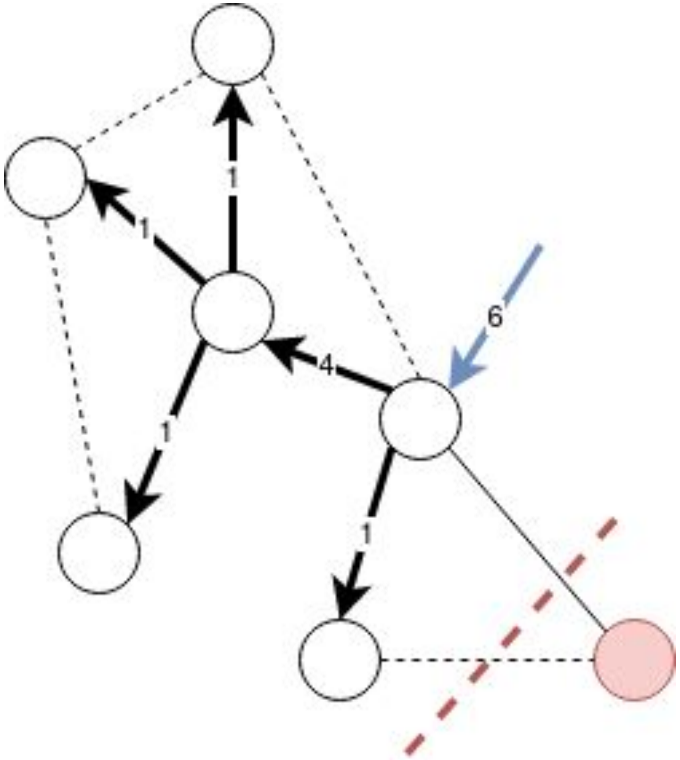
Prova do Teorema



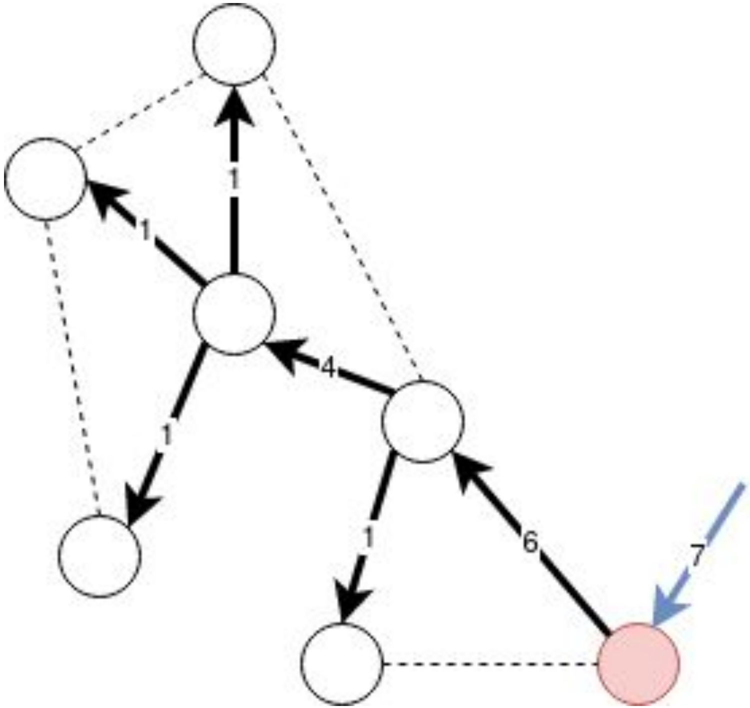
Prova do Teorema



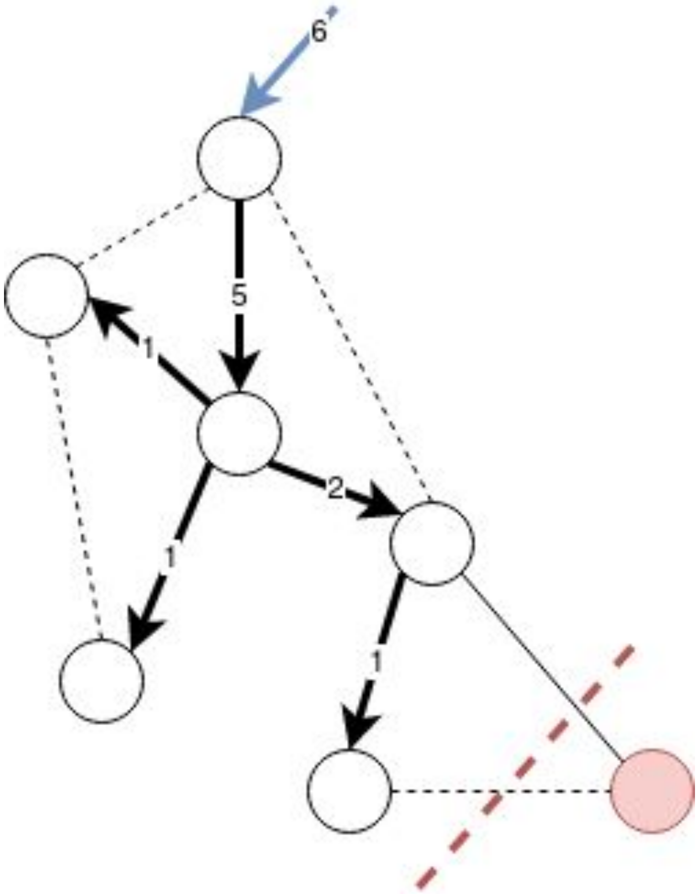
Prova do Teorema



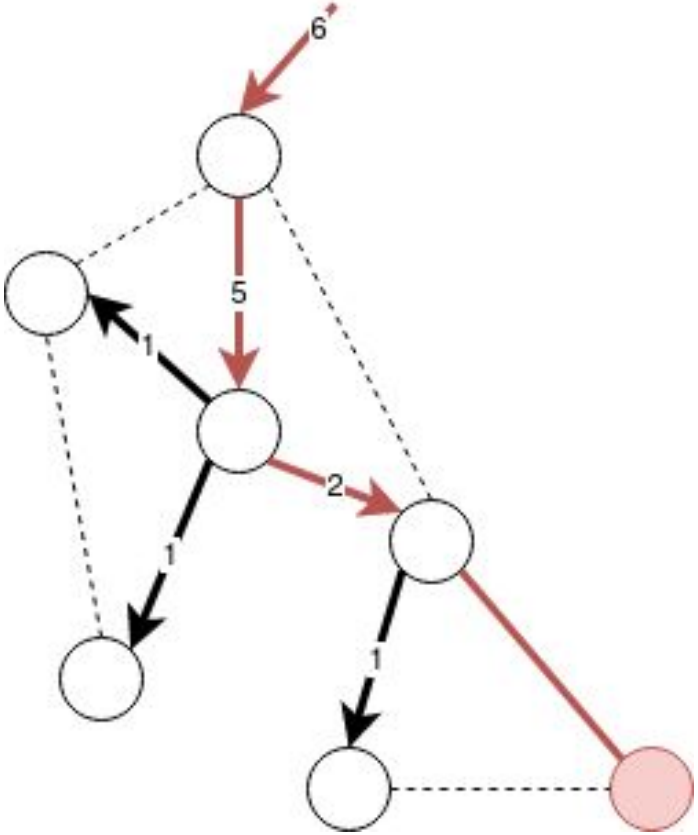
Prova do Teorema



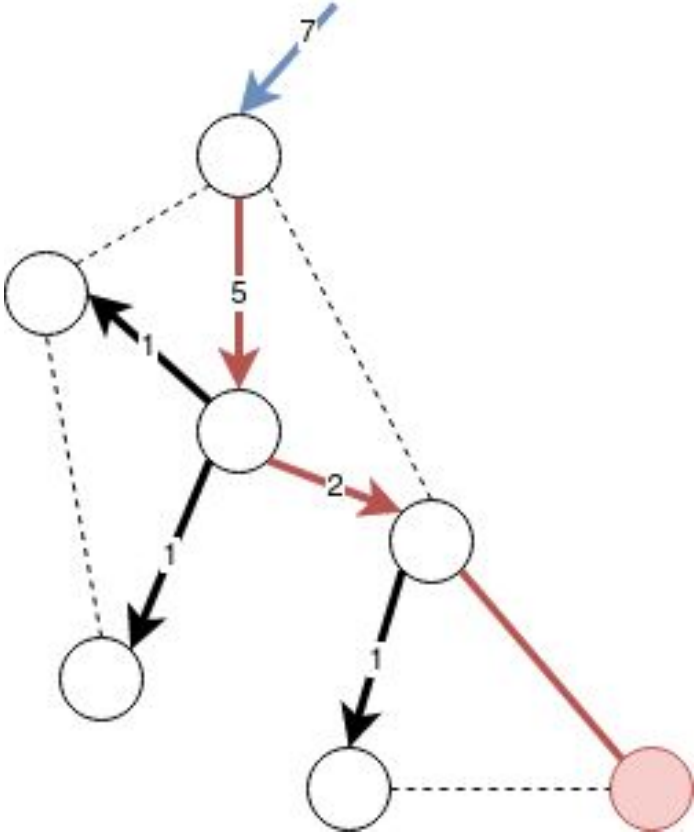
Prova do Teorema



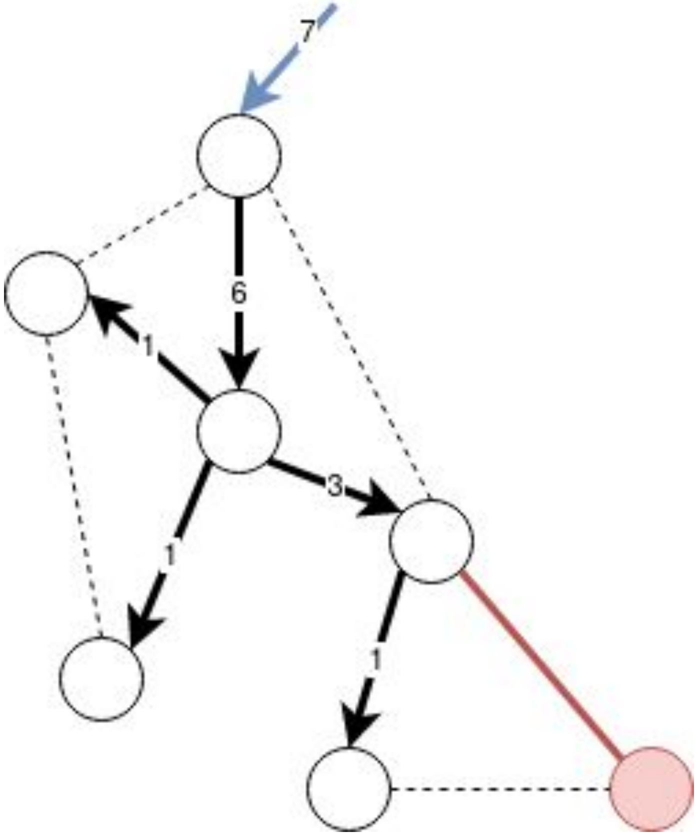
Prova do Teorema



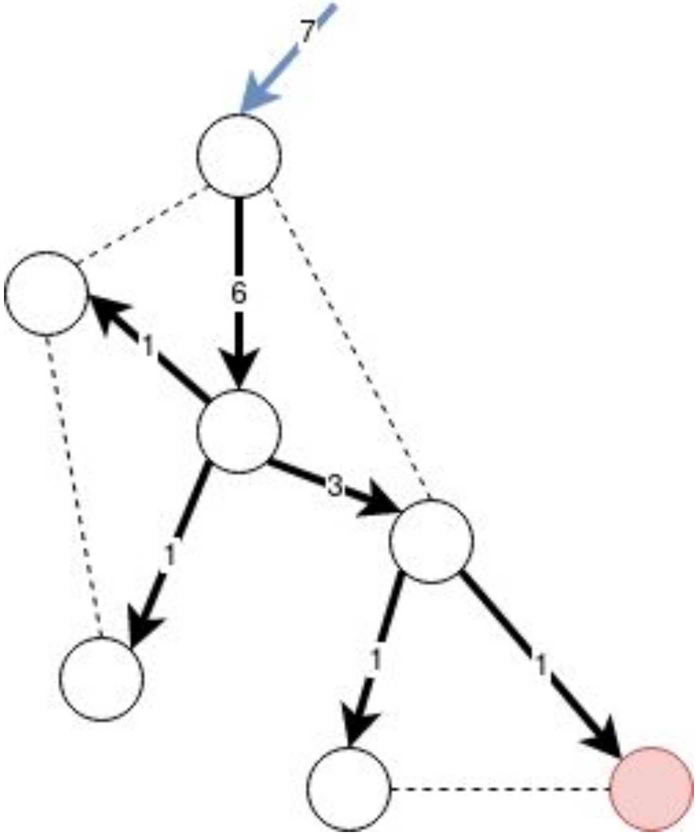
Prova do Teorema



Prova do Teorema



Prova do Teorema



Referências

- Franco, Á. J. P. Epidemic model with restricted circulation and social distancing on some network topologies. In: Cellular Automata. Cham: Springer International Publishing, 2021. p. 261–264.
- Carvalho C., Costa J., Sales C., Lopes R., Oliveira A. K. M. de, et al.. On the characterization of networks with multiple arc-disjoint branching flows. [Research Report] UFC; INRIA; CNRS; Université Côte d’Azur; I3S; LIRMM; Université de Montpellier. 2020.

Apoio

UNIVERSIDADE FEDERAL
DE SANTA CATARINA

ine
CTC • UFSC
Informática e estatística

fapesc
Fundação de Amparo à
Pesquisa e Inovação do
Estado de Santa Catarina

Obrigado!