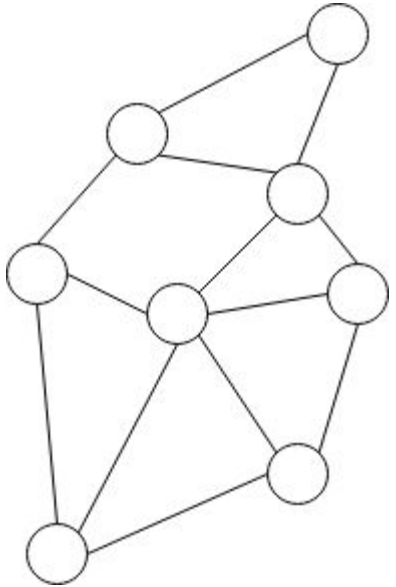
Minimização do Contágio em uma Topologia com Programação Matemática

Wesly C. Ataide, Álvaro J. P. Franco, Rafael de Santiago, Pedro B. Castellucci

O Nosso Problema

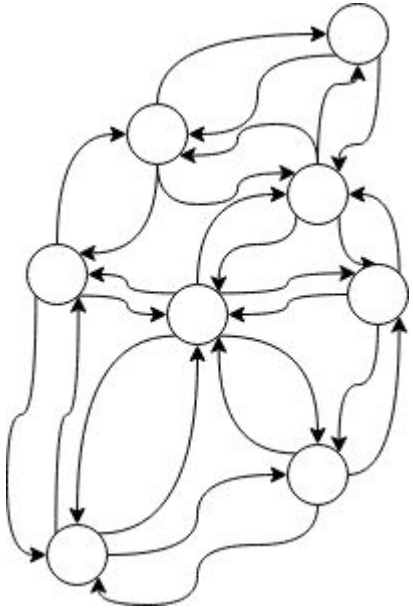
O Nosso Problema

- 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, chamados de valores beta.



O Nosso Problema

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

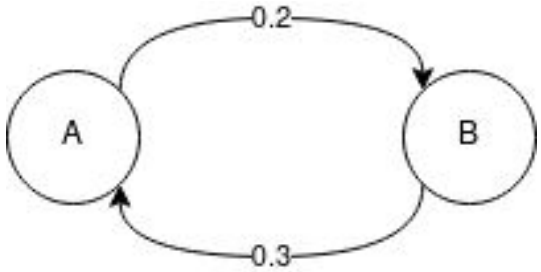
  A: S = 100; I = 10; R = 0;

  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**

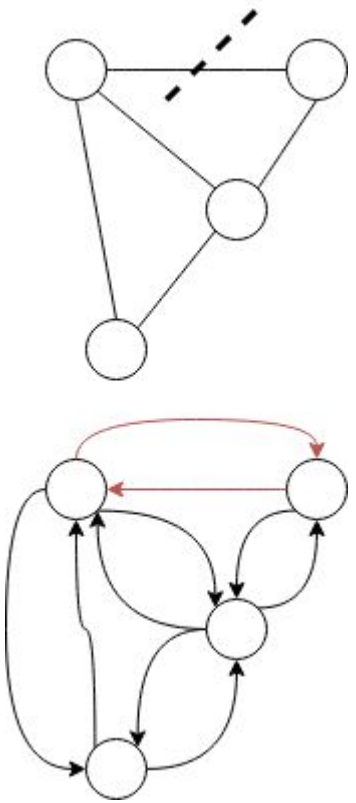
O Nosso Problema

- Probabilidade de um encontro com um infectado em um vértice é a proporção de infectados na população circulando

$$\mathbb{E}_i^t(\dot{S}_i^t, \dot{I}_i^t, \dot{R}_i^t) = \frac{S_{i\to i}\dot{I}_i^t}{\dot{S}_i^t + \dot{I}_i^t + \dot{R}_i^t} + \sum_{j\in\chi(i)} \frac{S_{i\to j}\dot{I}_j^t}{\dot{S}_j^t + \dot{I}_j^t + \dot{R}_j^t};$$

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!**



Modelo e implementação

Modelo e implementação

- Para cada arco (i, j), criamos uma variável binária **x** com valor 0 representando que a estrada (i, j) está bloqueada, e com valor 1 para representar que ela está livre.
- Definimos as vias como simétricas, ou seja, se (i, j) está liberada, (j, i) também.

- Cada arco também tem o valor beta associado.

Modelo e implementação

- Para cada escolha das variáveis **x**, acabamos com um certo valor de infectados.
- A nossa função objetivo, que tentamos minimizar, é o cálculo dos infectados para uma escolha das variáveis **x**.

$$\mathbb{E}_i^t(\dot{S}_i^t, \dot{I}_i^t, \dot{R}_i^t) = \frac{S_{i\to i}\dot{I}_i^t}{\dot{S}_i^t + \dot{I}_i^t + \dot{R}_i^t} + \sum_{j\in\chi(i)} \frac{S_{i\to j}\dot{I}_j^t}{\dot{S}_j^t + \dot{I}_j^t + \dot{R}_j^t};$$

Modelo e implementação

- Solver SCIP através da interface JuMP, uma linguagem para definir modelos de otimização embutida em Julia

- Tentamos minimizar o número de infectados para o próximo tempo

Modelo e implementação

- Como garantir a conexidade com restrições lineares ou não-lineares?

Modelo e implementação

- 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

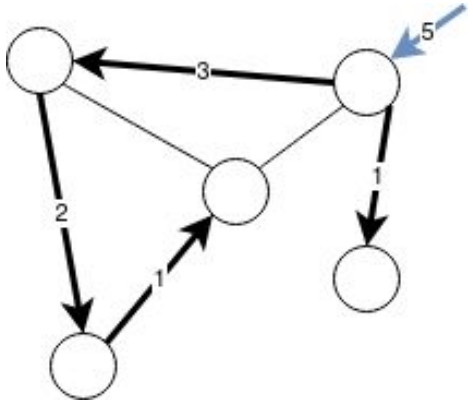
Duas condições:

- 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 induz a árvore T.

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

Modelo e implementação

- Para cada arco (i, j), adicionamos uma variável **f** correspondente.

- As restrições são análogas às condições do fluxo s-arborescente.
- Além disso, só deve ser permitido passar fluxo por um arco (i, j) se a variável **x** correspondente for igual a 1.

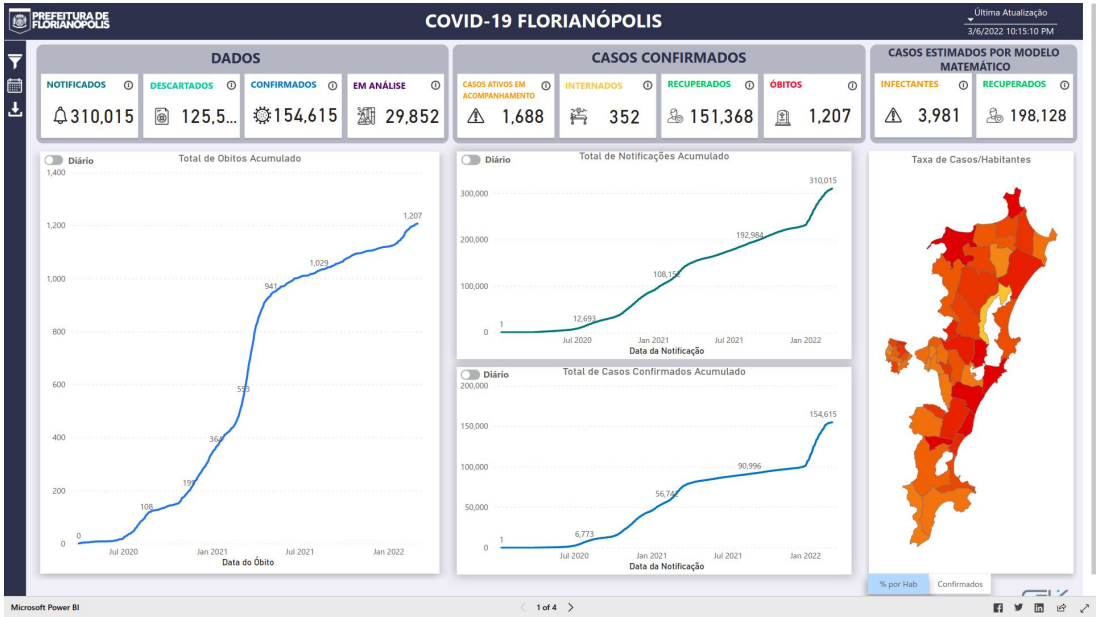
- Veja que isso não força a solução a ser uma árvore.

Construção de uma instância

Construção de uma instância

- Grafo baseado nos bairros de Florianópolis

- Dados da COVID-19 disponibilizados pela prefeitura
- Mas não temos a população

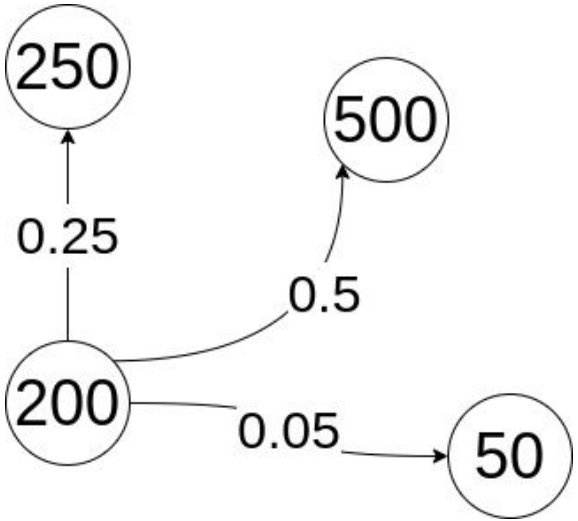


Construção de uma instância

- IBGE mantém estatísticas da população de Florianópolis, divididas por bairro
- Mas os bairros não correspondem diretamente
- Vários bairros agrupados em cada bairro do grafo anterior

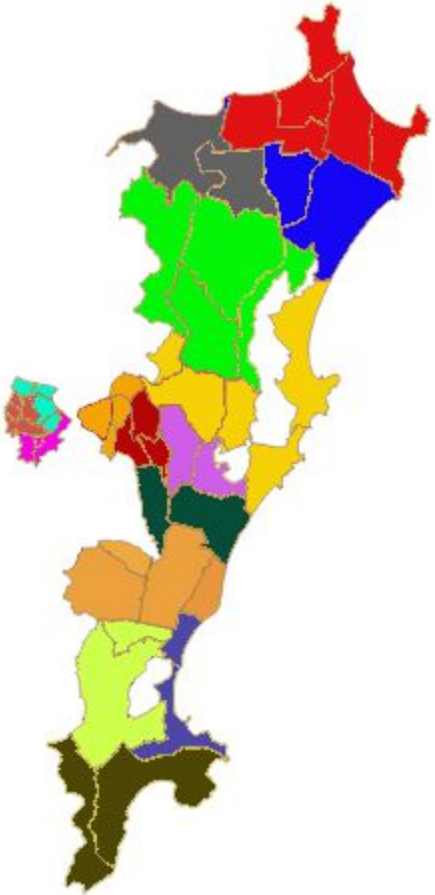
- Valores beta estimados

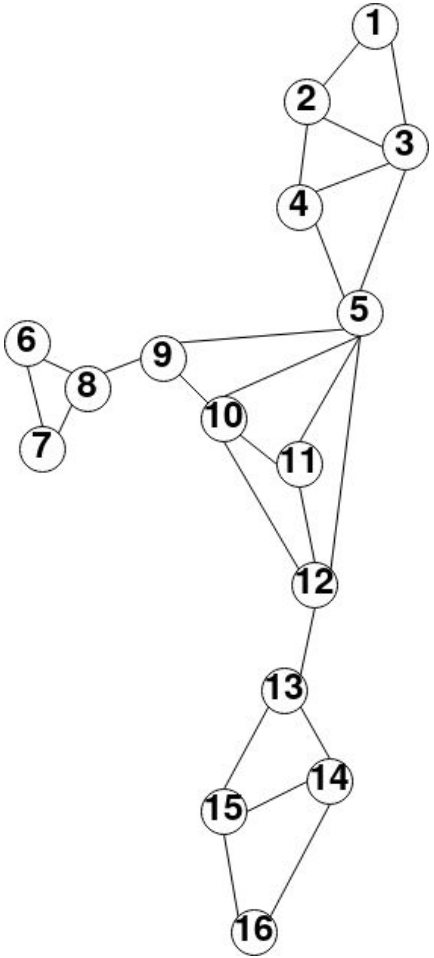
$$\beta_{(i,j)} = \frac{pop(j)}{pop(i) + \sum_{(i,k)} pop(k)}$$



Construção de uma instância

- Agrupar os bairros em regiões maiores (diminuir o número de vértices)





Solução inicial

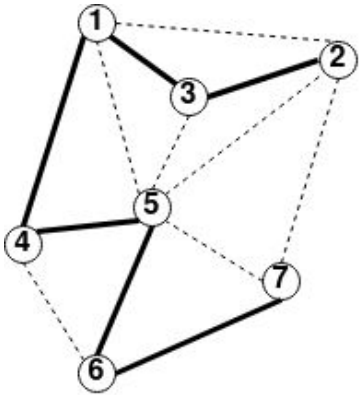
Solução inicial

- Vamos fornecer ao nosso otimizador uma solução viável (não necessariamente uma ótima)

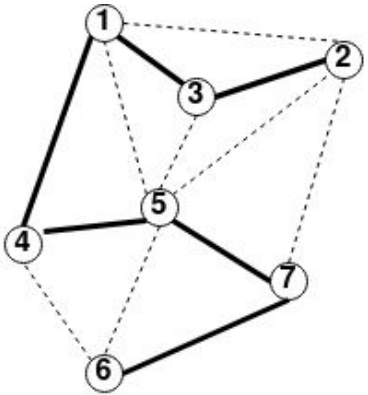
- A melhor árvore geradora que conseguirmos achar

Solução inicial

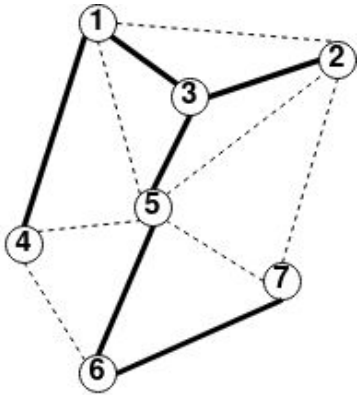
- Árvores adjacentes



(A)



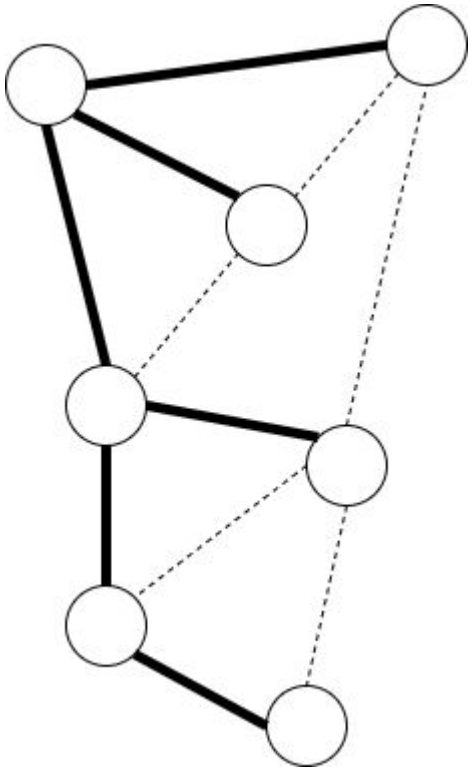
(B)



(C)

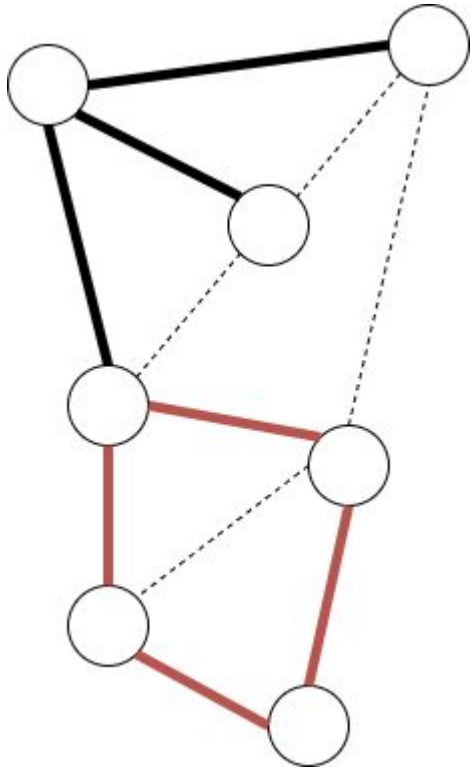
Solução inicial

- Começamos com uma árvore geradora qualquer
- Escolhemos uma aresta do grafo que não está na árvore e a adicionamos
- Testamos retirar cada uma das arestas do ciclo formado
- Repetimos para toda aresta não usada



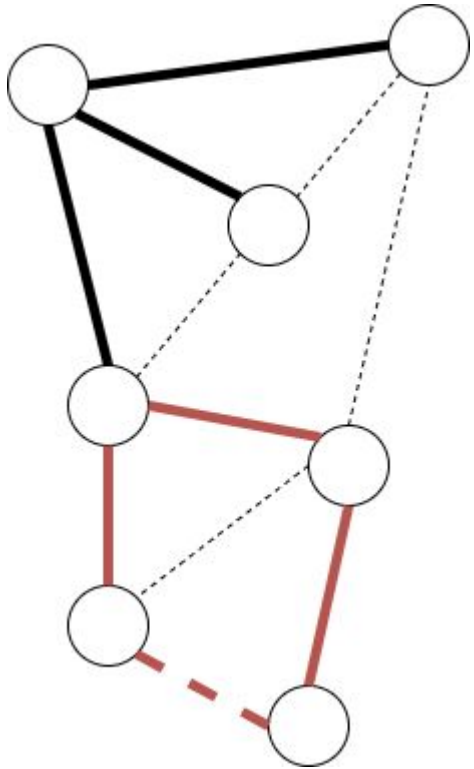
Solução inicial

- Começamos com uma árvore geradora qualquer
- Escolhemos uma aresta do grafo que não está na árvore e a adicionamos
- Testamos retirar cada uma das arestas do ciclo formado
- Repetimos para toda aresta não usada



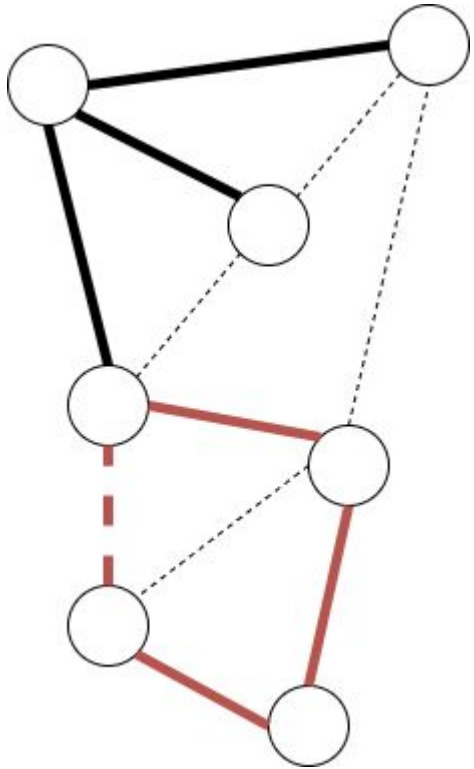
Solução inicial

- Começamos com uma árvore geradora qualquer
- Escolhemos uma aresta do grafo que não está na árvore e a adicionamos
- Testamos retirar cada uma das arestas do ciclo formado
- Repetimos para toda aresta não usada



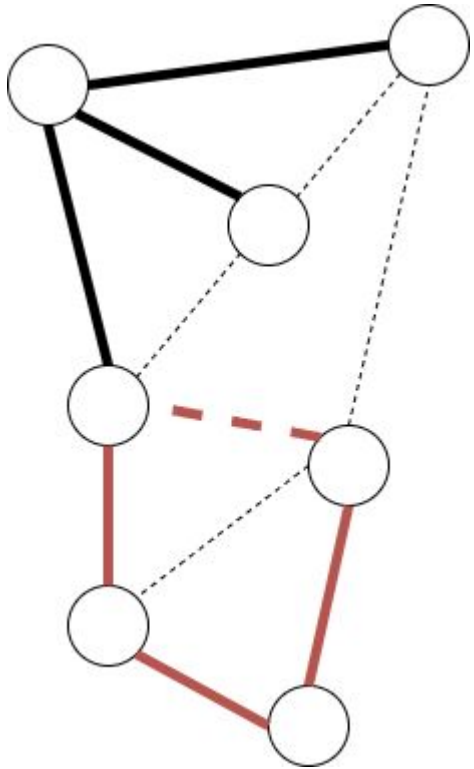
Solução inicial

- Começamos com uma árvore geradora qualquer
- Escolhemos uma aresta do grafo que não está na árvore e a adicionamos
- Testamos retirar cada uma das arestas do ciclo formado
- Repetimos para toda aresta não usada



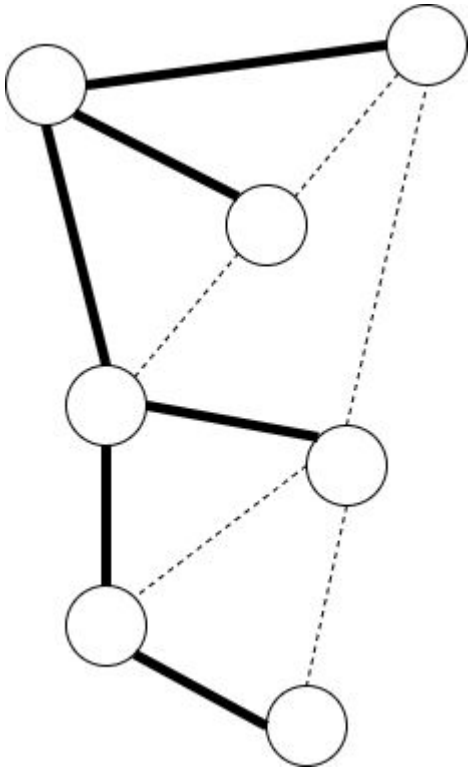
Solução inicial

- Começamos com uma árvore geradora qualquer
- Escolhemos uma aresta do grafo que não está na árvore e a adicionamos
- Testamos retirar cada uma das arestas do ciclo formado
- Repetimos para toda aresta não usada



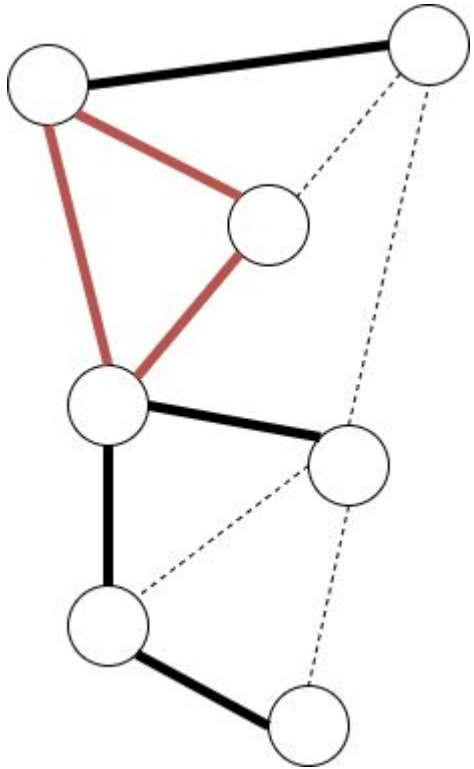
Solução inicial

- Começamos com uma árvore geradora qualquer
- Escolhemos uma aresta do grafo que não está na árvore e a adicionamos
- Testamos retirar cada uma das arestas do ciclo formado
- Repetimos para toda aresta não usada



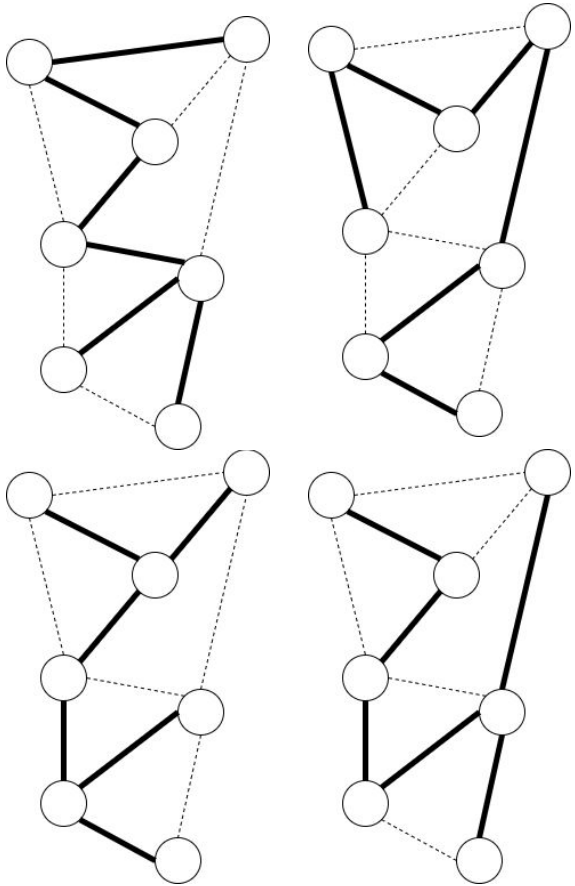
Solução inicial

- Começamos com uma árvore geradora qualquer
- Escolhemos uma aresta do grafo que não está na árvore e a adicionamos
- Testamos retirar cada uma das arestas do ciclo formado
- Repetimos para toda aresta não usada



Busca Iterada

- Mantemos sempre a com o menor número de infectados
- Repetimos esse processo para a árvore resultante
- Podemos reiniciar para árvores diferentes e conseguir árvores diferentes de resultado



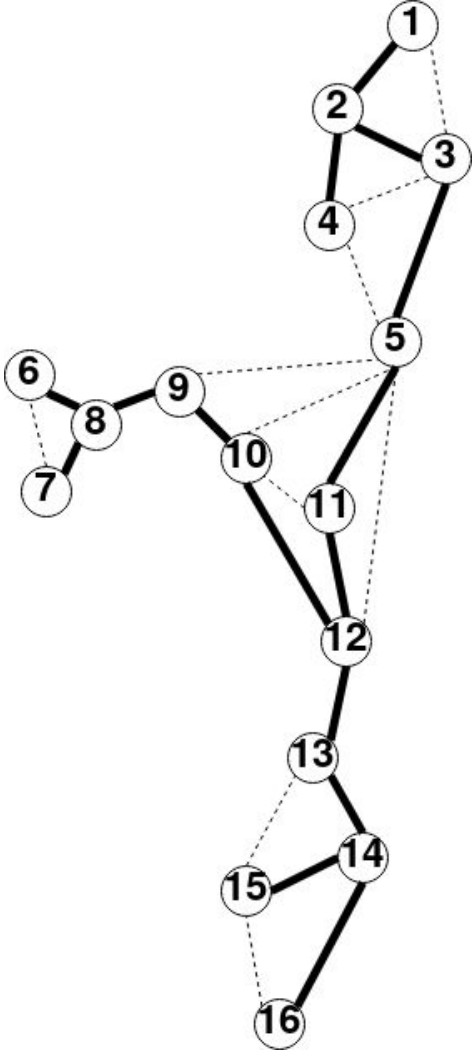
Resultados

Resultados: grafo de 16 regiões

- Resultado obtido da busca:

  (número esperado de infectados)

  ~622.2791



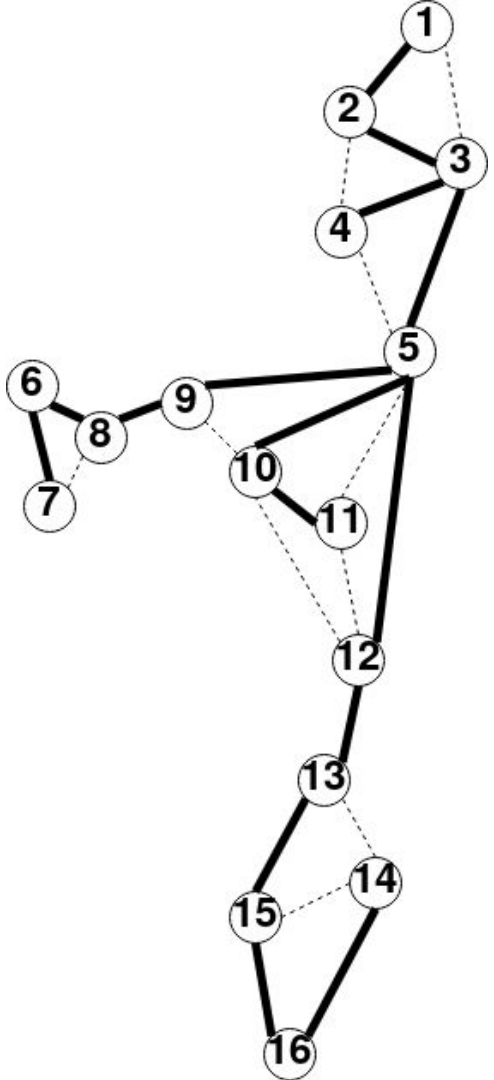
Resultados: grafo de 16 regiões

- Output do Julia:

  (número esperado de infectados)

  ~622.2387

- Mas esse valor não é preciso



Trabalhos futuros

- Implementação em outras ferramentas

- Revisão do modelo

- Investigação sobre a possível linearização do modelo
  - Não-linearidade da função objetivo

Obrigado pela atenção!

Agradecimentos:

CENTRO TECNOLÓGICO
Universidade Federal de Santa Catarina

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

UFSC
PROPESQ

Obrigado pela atenção!

Agradecimentos:

CENTRO TECNOLÓGICO
Universidade Federal de Santa Catarina

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

UFSC
PROPESQ

$$
\begin{array}{lrcl}
\min & \sum_{i \in V} I_i^{t+1} & & \\
\text{sujeito a:} & \dot{S}_i^t & = & S_{i \to i}^t + \sum_{j \in \chi(i)} S_{j \to i}^t, \forall i \in V \\
& \dot{I}_i^t & = & I_{i \to i}^t + \sum_{j \in \chi(i)} I_{j \to i}^t, \forall i \in V \\
& \dot{R}_i^t & = & R_{i \to i}^t + \sum_{j \in \chi(i)} R_{j \to i}^t, \forall i \in V \\
& \sum_{(i,j) \in E} f_{i,j} + 1 & = & \sum_{(j,i) \in E} f_{j,i}, \forall i \in V \text{ and } i \neq s \\
& \sum_{(s,j) \in E} f_{s,j} & = & n - 1 \\
& n x_{i,j} & \geq & f_{i,j}, \forall (i,j) \in E \\
& f_{i,j} & \in & \{0, \ldots, n-1\}, \forall (i,j) \in E \\
& x_{i,j} & \in & \{0,1\}, \forall (i,j) \in E,
\end{array}
$$

onde

$$I_i^{t+1} = I_i^t - \xi I_i^t + v \mathbb{E}_i^t(\dot{S}_i^t, \dot{I}_i^t, \dot{R}_i^t);$$

$$\mathbb{E}_i^t(\dot{S}_i^t, \dot{I}_i^t, \dot{R}_i^t) = \frac{S_{i \to i} \dot{I}_i^t}{\dot{S}_i^t + \dot{I}_i^t + \dot{R}_i^t} + \sum_{j \in \chi(i)} \frac{S_{i \to j} \dot{I}_j^t}{\dot{S}_j^t + \dot{I}_j^t + \dot{R}_j^t};$$

$$S_{i \to i}^t = (1 - \sum_{j \in \chi(i)} (\beta_{i,j} x_{i,j}^t)) S_i^t;\ I_{i \to i}^t = (1 - \sum_{j \in \chi(i)} (\beta_{i,j} x_{i,j}^t)) I_i^t;\ R_{i \to i}^t = (1 - \sum_{j \in \chi(i)} (\beta_{i,j} x_{i,j}^t)) R_i^t;$$

$$S_{j \to i}^t = \beta_{j,i} x_{j,i}^t S_j^t;\ I_{j \to i}^t = \beta_{j,i} x_{j,i}^t I_j^t;\ R_{j \to i}^t = \beta_{j,i} x_{j,i}^t R_j^t;$$