

**UNIVERSIDADE FEDERAL DE LAVRAS**

Ciência da Computação

# **SOLUÇÃO PARA O PROBLEMA DAS FAKE NEWS**

**Heitor Rodrigues Sabino  
Ranulfo Mascari Neto**

Trabalho Prático Final  
GCC118 - Programação Matemática

Professora: **Andreza C. Beezão Moreira**  
Professor: **Mayron César O. Moreira**

**Lavras - MG  
2024**

# 1 Introdução

A disseminação de notícias falsas, ou popularmente conhecidas como *fake news*, é um problema gravíssimo que está literalmente ao alcance dos nossos dedos. De acordo com uma matéria publicada no *Journal* da USP [1], entre 21 países pesquisados, o Brasil possui a população com maior dificuldade em identificar *fake news*.

Uma forma de coibir que tais notícias sejam propagadas é criar mecanismos de atraso na difusão de pacotes de dados pouco confiáveis nas redes de computadores. Para tanto, suponha que uma mensagem classificada como suspeita por meio de uma rede neural seja enviada através de um vértice  $s \in V$  de uma rede modelada por um grafo direcionado  $G(V, E)$ . O conjunto de vértices  $V$  corresponde aos servidores que processam as informações, enquanto os arcos  $(u, v) \in E$  são os enlaces de comunicação entre os servidores  $u, v$ . Deseja-se reduzir o número de servidores em que a notícia falsa pode se encontrar após um período de tempo igual a  $T$ .

A propagação de uma informação no enlace  $(u, v) \in E$  possui tempo estimado igual a  $t_{uv}$ . O Setor de Tecnologia da Informação (STI) do governo federal possui recursos (softwares auxiliares, por exemplo) tecnológicos que podem ser aplicados aos servidores para retardar o tempo de broadcast da *fake news*. Assim, caso o servidor  $u$  possua algum dos  $\alpha$  recursos instalados, o tempo de transmissão da *fake news* no enlace  $(u, v)$  seria  $t_{uv} + \delta$ , em que  $\delta > 0$  representa um atraso de propagação. Cada um dos  $i = 1, \dots, \alpha$  recursos fica disponível no instante  $\beta_i$ .

A solução do problema consiste em determinar a designação dos recursos aos servidores. Cada servidor receberá, no máximo, um recurso. Considera-se que um recurso  $i$  é alocado no menor tempo  $\beta_i$ . O objetivo definido é a minimização da quantidade de servidores em que a *fake news* poderia chegar em um tempo menor ou igual a  $T$ .

## 2 Modelagem do Problema

A disseminação de fake news é um problema significativo, sendo um desafio identificar e conter a propagação dessas informações nas redes digitais. A modelagem proposta visa minimizar o número de servidores infectados por uma notícia falsa ao longo do tempo, considerando a possibilidade de aplicação de recursos para retardar sua propagação.

### 2.1 Definição do Problema

Seja um grafo direcionado  $G(V, E)$ , onde:

- $V$  representa os servidores;
- $E$  representa os canais de comunicação entre servidores;
- $t_{uv}$  denota o tempo necessário para a propagação da fake news através do arco  $(u, v) \in E$ ;
- Um conjunto de recursos limitados pode ser alocado a servidores para aumentar o tempo de propagação em um valor  $\delta$ ;
- $T$  é o tempo limite para considerar a propagação;
- O objetivo é minimizar o número total de servidores infectados considerando todos os possíveis pontos iniciais da fake news.

## 2.2 Parâmetros

- $G = (V, E)$ : grafo representando a rede de servidores;
- $t_{uv}$ : tempo normal de propagação pelo arco  $(u, v)$ ;
- $\delta$ : atraso causado por um recurso alocado;
- $\beta_i$ : tempo mínimo em que o recurso  $i$  pode ser alocado;
- $T$ : tempo limite para propagação;
- $\alpha$ : número máximo de recursos disponíveis.

## 2.3 Variáveis de Decisão

- $y_{v,s} \in \{0, 1\}$ : indica se o servidor  $v$  é infectado quando a fake news começa em  $s$ ;
- $z_{uv,s} \in \{0, 1\}$ : indica se a fake news propaga de  $u$  para  $v$  quando começa em  $s$ ;
- $t_{v,s} \geq 0$ : tempo em que o servidor  $v$  é infectado quando a fake news começa em  $s$ ;
- $x_v \in \{0, 1\}$ : indica se um recurso é alocado no servidor  $v$ ;
- $w_{vi} \in \{0, 1\}$ : indica se o recurso  $i$  é alocado ao servidor  $v$ .

## 2.4 Função Objetivo

O objetivo é minimizar o número total de infecções considerando todos os pontos iniciais da fake news:

$$\min \sum_{s \in V} \sum_{v \in V} y_{v,s} \quad (1)$$

## 2.5 Restrições

**1. Condição inicial de infecção** Se a fake news começa em  $s$ , então  $s$  está infectado no instante 0:

$$y_{s,s} = 1, \quad \forall s \in V \quad (2)$$

$$t_{s,s} = 0, \quad \forall s \in V \quad (3)$$

**2. Propagação da fake news** A fake news pode se propagar de  $u$  para  $v$  somente se  $u$  já estiver infectado:

$$z_{uv,s} \leq y_{u,s}, \quad \forall (u, v) \in E, \forall s \in V \quad (4)$$

Se a fake news se propaga de  $u$  para  $v$ , então  $v$  deve ser infectado:

$$y_{v,s} \geq z_{uv,s}, \quad \forall (u, v) \in E, \forall s \in V \quad (5)$$

**3. Tempo de propagação considerando os recursos** O tempo mínimo de infecção do servidor  $v$  depende do tempo de propagação  $t_{uv}$  e do atraso  $\delta$  se  $x_u = 1$ :

$$t_{v,s} \geq t_{u,s} + t_{uv} + x_u \cdot \delta, \quad \forall (u, v) \in E, \forall s \in V \quad (6)$$

Para garantir que um servidor  $v$  não seja infectado se seu tempo de infecção ultrapassar  $T$ , introduzimos uma variável binária auxiliar  $\gamma_{v,s}$ :

$$y_{v,s} \leq M \cdot (1 - \gamma_{v,s}), \quad \forall v \in V, \forall s \in V \quad (7)$$

$$t_{v,s} \leq T + M \cdot \gamma_{v,s}, \quad \forall v \in V, \forall s \in V \quad (8)$$

onde  $M$  é um valor suficientemente grande.

**4. Restrição de alocação de recursos** Cada servidor pode receber no máximo um recurso, e o total de recursos distribuídos não pode ultrapassar  $\alpha$ :

$$\sum_{v \in V} x_v \leq \alpha \quad (9)$$

**5. Disponibilidade dos recursos** Um servidor  $v$  só pode receber um recurso se houver pelo menos um disponível no tempo certo  $\beta_i$ :

$$t_{v,s} \geq \sum_{i \in I} \beta_i \cdot w_{vi}, \quad \forall v \in V, \forall s \in V \quad (10)$$

**6. Consistência da alocação de recursos** Se um servidor recebe um recurso, ele deve estar corretamente alocado:

$$x_v = \sum_{i \in I} w_{vi}, \quad \forall v \in V \quad (11)$$

### 3 Descrição da Solução

A solução para o problema de minimização da propagação de \*fake news\* foi implementada utilizando a biblioteca Gurobi para resolver um modelo de otimização baseado em programação inteira mista. O código foi estruturado para modelar a propagação da informação como um problema de fluxo em um grafo direcionado, onde os servidores são representados pelos nós e as conexões entre eles pelas arestas.

O modelo define variáveis de decisão que representam a infecção dos servidores ao longo do tempo, a propagação da fake news entre conexões e a alocação de recursos para retardar essa propagação. A função objetivo busca minimizar o número total de servidores infectados, enquanto as restrições garantem que a propagação respeite os tempos de transmissão e os atrasos impostos pelos recursos.

O código lê uma instância do problema contendo a estrutura do grafo e as especificações dos recursos disponíveis. A seguir, o modelo é criado utilizando o solver Gurobi, definindo variáveis binárias para representar infecções e alocações de recursos, além de variáveis contínuas para medir o tempo de infecção. As restrições incluem:

- A infecção inicial do servidor fonte em tempo zero;
- A propagação da infecção somente se o servidor anterior estiver infectado;

- O tempo de infecção respeitando os atrasos impostos pelos recursos;
- A restrição de que um servidor pode receber no máximo um recurso;
- A limitação no número total de recursos alocados.

Após a formulação do modelo, o solver busca a solução ótima minimizando a propagação da fake new. A saída do código exibe os servidores infectados em cada cenário, o valor ótimo encontrado e o melhor cenário identificado, onde a infecção é minimizada. Esse modelo permite avaliar diferentes estratégias de mitigação, ajudando a tomar decisões mais eficazes no controle da disseminação de informações falsas em redes computacionais.

### 3.1 Simulated Annealing

O algoritmo *Simulated Annealing* (SA) é uma meta-heurística inspirada no processo de recozimento metalúrgico, no qual um material é aquecido a altas temperaturas e gradualmente resfriado para atingir uma configuração de baixa energia. Essa analogia é usada para ilustrar a ideia de que o algoritmo começa com uma solução aleatória e vai refinando essa solução com o tempo, em busca de uma solução ótima ou suficientemente boa.

O processo de *Simulated Annealing* funciona da seguinte forma:

1. **Solução Inicial:** O algoritmo começa com uma solução inicial aleatória. Essa solução pode não ser a melhor, mas é um ponto de partida.
2. **Temperatura Inicial:** O processo começa com uma temperatura alta. A temperatura controla a probabilidade de aceitar uma solução pior que a atual. Em um primeiro momento, com a temperatura alta, o algoritmo tem maior liberdade para explorar o espaço de soluções, aceitando soluções que aumentem o custo (ou pioram a solução).
3. **Procurar Soluções Vizinhas:** O algoritmo gera uma solução vizinha, que é uma pequena variação da solução atual. A vizinhança é determinada por um conjunto de operações de modificação (ex.: troca de valores, deslocamento de elementos, etc.).
4. **Aceitação de Soluções Piores:** Se a nova solução é melhor que a atual (ou seja, seu custo é menor), ela é aceita automaticamente. Se a nova solução é pior, ela pode ser aceita com uma certa probabilidade, que depende da temperatura e da diferença de custo entre as soluções. Quanto maior a temperatura, maior a chance de aceitar uma solução pior. Essa característica é importante para evitar que o algoritmo fique preso em mínimos locais.
5. **Resfriamento Gradual:** A cada iteração, a temperatura é reduzida de maneira controlada, geralmente de forma exponencial, até atingir um valor mínimo predefinido, o que diminui gradualmente a probabilidade de aceitar soluções piores. Isso ajuda a refinar a busca para encontrar uma solução mais otimizada ao longo do tempo.
6. **Critério de Parada:** O algoritmo continua até que a temperatura alcance um valor mínimo definido ( $T_{min}$ ) ou um número máximo de iterações seja atingido.

## 4 Resultados e Análise

### 4.1 Análise dos Resultados

A análise dos resultados foi realizada levando em consideração o impacto de diferentes números de iterações e temperaturas iniciais  $T_{ini}$  no desempenho da meta-heurística Simulated Annealing (SA).

---

**Algorithm 1** Simulated Annealing

---

```
1:  $s \leftarrow s_0$  {Geração da solução inicial}
2:  $T \leftarrow T_{ini}$  {Temperatura inicial}
3: repeat
4:   repeat
5:     Gerar um vizinho aleatório  $s'$ 
6:      $\Delta E \leftarrow \text{calcular\_custo}(s') - \text{calcular\_custo}(s)$ 
7:     if  $\Delta E < 0$  then
8:        $s \leftarrow s'$  {Aceita a solução vizinha}
9:     else
10:      Aceita  $s'$  com probabilidade  $e^{-\Delta E/T}$ 
11:    end if
12:  until Condição de equilíbrio
13:   $T \leftarrow \max(T_{min}, T \times \alpha)$  {Atualiza a temperatura}
14: until  $T < T_{min}$ 
15: output: Melhor solução encontrada
```

---

Table 1: Resultados para iterações = 100,  $T_{ini} = 200$ 

Instância	SI	SF	Desvio SI $\rightarrow$ SF	Desvio SF $\rightarrow$ Ótima	Tempo Médio (s)
fn2	269.00	279.80	-4.01%	47.26%	8.7626
fn3	282.00	293.60	-4.11%	41.84%	9.6107
fn1	281.00	294.00	-4.63%	55.56%	18.6748
fn5	299.00	311.60	-4.21%	37.88%	24.7800
fn6	281.00	291.80	-3.84%	48.88%	24.1733
fn7	263.00	278.80	-6.01%	42.24%	19.9965
fn8	280.00	293.40	-4.79%	37.75%	22.4983
fn9	295.00	308.60	-4.61%	36.55%	31.4228
fn10	297.00	310.80	-4.65%	32.26%	26.6980

Table 2: Comparação das Instância fn2 e fn3 para diferentes números de iterações

Iterações	Instância	SI	SF	Desvio SI $\rightarrow$ SF	Desvio SF $\rightarrow$ Ótima	Tempo Médio (s)
10	fn2	269.00	278.20	-3.42%	46.42%	0.9237
10	fn3	282.00	292.40	-3.69%	41.26%	1.0252
100	fn2	269.00	279.80	-4.01%	47.26%	8.7626
100	fn3	282.00	293.60	-4.11%	41.84%	9.6107
1000	fn2	269.00	261.60	2.75%	37.68%	85.3455
1000	fn3	282.00	279.00	1.06%	34.78%	95.8376

Table 3: Comparação das Instância fn2 e fn3 para diferentes valores de  $T_{ini}$ 

$T_{ini}$	Instância	SI	SF	Desvio SI $\rightarrow$ SF	Desvio SF $\rightarrow$ Ótima	Tempo Médio (s)
200	fn2	269.00	279.80	-4.01%	47.26%	8.7626
200	fn3	282.00	293.60	-4.11%	41.84%	9.6107
2000	fn2	269.00	279.20	-3.79%	46.95%	8.6235
2000	fn3	282.00	294.00	-4.26%	42.03%	9.4297

- **Efeito do Número de Iterações:** Como mostrado na Tabela 2, o aumento do número de iterações não levou necessariamente a uma melhora na solução encontrada. Para 10 iterações, os desvios foram menores, mas o tempo computacional também foi significativamente mais baixo, com valores de tempo médios de cerca de 1 segundo. Já para 100 e 1000 iterações, o desvio aumentou, mostrando que o algoritmo, ao ser forçado a explorar mais soluções, acabou se afastando das melhores soluções. O tempo de execução também aumentou consideravelmente, passando de cerca de 8 segundos para mais de 80 segundos.
- **Efeito da Temperatura Inicial  $T_{ini}$ :** Conforme mostrado na Tabela 3, não houve uma grande diferença nos resultados ao alterar a temperatura inicial  $T_{ini}$  de 200 para 2000. O desvio ficou em uma faixa de -4.01% a -4.11% para  $T_{ini} = 200$  e de -3.79% a -4.26% para  $T_{ini} = 2000$ , com tempos de execução muito próximos (em torno de 8 a 9 segundos). Isso sugere que a temperatura inicial teve um impacto limitado nos resultados gerais da meta-heurística.
- **Desempenho das Instâncias:** As instâncias  $fn2$  e  $fn3$  apresentaram comportamentos similares em termos de resultados. Para ambas, o desvio foi baixo e o tempo médio de execução foi relativamente estável entre diferentes números de iterações e valores de temperatura inicial. No entanto, a instância  $fn3$  apresentou um desvio um pouco maior em comparação com  $fn2$ , especialmente com um número de iterações maior (1000), sugerindo que o problema de  $fn3$  pode ser um pouco mais complexo para o algoritmo.

## 5 Conclusão

Com base nos resultados obtidos, pode-se concluir que a meta-heurística Simulated Annealing foi eficaz na obtenção de boas soluções para o problema de disseminação de fake news em redes de computadores. No entanto, foi observado que o aumento do número de iterações não levou necessariamente a uma melhora nas soluções encontradas, uma vez que, ao forçar o algoritmo a explorar mais soluções, ele acabou se afastando das melhores soluções. Além disso, o tempo de execução aumentou consideravelmente com o aumento das iterações, tornando o processo mais lento sem uma melhora significativa na qualidade da solução. A temperatura inicial  $T_{ini}$  teve um impacto limitado nos resultados gerais, com desvios próximos entre os valores de 200 e 2000. A análise das instâncias  $fn2$  e  $fn3$  revelou comportamentos semelhantes, embora a instância  $fn3$  tenha mostrado um desvio ligeiramente maior, indicando que ela pode ser mais complexa para o algoritmo. Em termos de parâmetros ideais, um número intermediário de iterações e uma temperatura inicial moderada são os mais indicados para equilibrar a qualidade da solução e o tempo computacional.

## 6 Bibliografia

### References

- [1] JORNAL DA USP. Relatório da OCDE mostra que brasileiros são os piores em identificar notícias falsas. Disponível em: <https://jornal.usp.br/radio-usp/>

relatorio-da-ocde-mostra-que-brasileiros-sao-os-piores-em-identificar-noticias-  
Acesso em: 2025.