

# **Uso de Redes Neurais em Grafos para Otimização de Tempo no Algoritmo IMOPSO de Maximização de Influência**

**Ana Luiza Almeida Soares<sup>1</sup>, Rodrigo César Pedrosa Silva<sup>1</sup>**

<sup>1</sup>Programa de Pós-Graduação em Ciência da Computação

Departamento de Computação

Universidade Federal de Ouro Preto

ana.almeida3@aluno.ufop.edu.br, rodrigo.silva@ufop.edu.br

## **1. Introdução**

O problema de Maximização de Influência busca identificar o conjunto de pessoas que maximiza a propagação de influência em uma rede social. Em outras palavras, considerando uma rede social onde os nós representam pessoas e as arestas indicam o nível de influência que uma pessoa exerce sobre outra, o objetivo é encontrar o conjunto de nós que maximize o número de outros nós influenciados.

Esse problema possui inúmeras aplicações, desde a divulgação de produtos em redes sociais até a análise da eficácia da disseminação de informações políticas. O campo vem sendo amplamente estudado devido ao impacto significativo que as redes sociais digitais exercem na formação da opinião pública.

A solução ideal para o problema envolveria avaliar todas as permutações possíveis de conjuntos, mas isso é inviável para redes grandes, devido ao tempo necessário para construir e comparar cada permutação em busca da melhor.

Consequentemente, várias abordagens foram desenvolvidas. A maioria dessas soluções considera o tamanho do conjunto de influenciadores como um valor fixo, ou seja, a quantidade de pessoas escolhidas para influenciar é uma entrada do algoritmo. Essa abordagem, no entanto, limita a aplicação dessas soluções, já que nem sempre é possível determinar previamente quantos nós devem ser escolhidos, sendo o custo da escolha desses nós, em geral, o principal limitador. Para contornar essa limitação, algumas soluções permitem que o tamanho do conjunto seja variável, impondo uma restrição de custo que se adequa a um orçamento específico, passado como parâmetro ao modelo.

Uma dessas soluções é o algoritmo IMOPSO, que utiliza otimização por enxame de partículas (swarm optimization) e busca local para encontrar uma solução próxima da ótima. Como esse algoritmo emprega heurísticas, ele não garante a solução exata, mas tem apresentado excelentes resultados, especialmente para redes maiores, com uma considerável economia de tempo.

O projeto propõe uma melhoria nesse modelo ao introduzir uma etapa de pré-seleção de nós relevantes para a avaliação. Esses nós serão identificados por uma rede neural de grafos (Graph Neural Network, GNN) com base nas informações derivadas de alguns caminhos mínimos na rede. Essa abordagem busca reduzir o tempo de execução do algoritmo IMOPSO, mantendo sua precisão na solução do problema de Maximização de Influência.

como calcular a influência

2. Materiais e Métodos

Tecnologias, instrumentos e procedimentos que serão usados no estudo. O Algoritmo 1 se refere ao método de ordenação Bubblesort expresso em linguagem Python.

Algoritmo 1. Método de ordenação Bubblesort

```
1 def bubble_sort(alist):
2     for i in range(len(alist)-1,0,-1):
3         for j in range(i):
4             if alist[i]>alist[i+1]:
5                 temp = alist[i]
6                 alist[i] = alist[i+1]
7                 alist[i+1] = temp
```



Figura 1. Minha figura

Tabela 1. Minha tabela

cabeçalho 1	cabeçalho 2
texto à esquerda	Existem muitas variações das passagens do Lorem Ipsum disponíveis, mas a maior parte sofreu alterações de alguma forma, pela injeção de humor, ou de palavras aleatórias que nem sequer parecem suficientemente credíveis. Se vai usar uma passagem do Lorem Ipsum, deve ter a certeza que não contém nada de embaraçoso escondido no meio do texto.

3. Resultados Esperados

Essa seção deverá ser escrita na segunda parte do trabalho, conhecida como TCC2.