Problema do Conjunto Dominante

Paulo Victor Pires Siqueira, Erick dos Santos Batista

Universidade Federal de Roraima (UFRR)

Boa Vista - RR - Brasil

psiqueira.paulo@gmail.com, dsantoserick@gmail.com

Resumo. Este relatório vem a complementar o trabalho apresentado em sala de aula, onde colocamos em prática o uso do algoritmo exato de conjuntos dominantes, junto a uma melhoria usada para solucionar um problema proposto, trazendo assim estudos sobre os resultados, os algoritmos foram desenvolvidos usando linguagem C e C++ nas aplicações.

1. Introdução

Este projeto foi desenvolvido com o objetivo de desenvolver e apresentar o algoritmo para o Problema do Conjunto Dominante, tanto em sua forma exata quanto como um algoritmo aproximado usando técnicas de otimização, além de pôr em prática usando uma aplicação real. Ao final são realizados testes para confirmar o melhoramento obtido.

2. Definição do Problema

Descrevendo da maneira mais formal possível, dado um grafo G = (V,E), um conjunto dominante é um subconjunto D de V no qual cada vértice do grafo G está contido em D ou é adjacente a um vértice de D. O número de dominação ɣ(G) representa a quantidade de vértices contidos em no conjunto dominante. O objetivo do problema é encontrar o menor valor para ɣ(G), de modo que satisfaça as condições básicas.

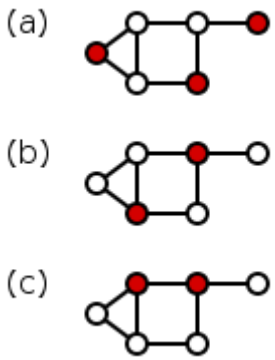

Figura 1: Exemplos de Conjuntos Dominantes (Vértices Vermelhos)

O problema pode ter duas configurações diferentes como resultado, isto é, existe uma subcategoria do problema denominado Conjunto Dominante Conectado e sua principal característica é garantir que os vértices do conjunto seja adjacente a pelo menos um outro vértice dominante.

3. Descrição do Caso de Aplicação

Um exemplo aplicação do problema do conjunto dominante consiste em distribuir sensores autônomos por um determinado espaço para monitorar condições físicas e ambientais. É preciso que haja comunicação entre os sensores, porém assim como os computadores eles são suscetíveis a ataques ou falhas, por isso é importante a criação de um backbone desses sensores, ou seja, selecionar sensores que fazem conexão com o máximo de outros sensores e utilizar os mesmos como meio de comunicação entre o sensores fisicamente distantes. Logo esses sensores intermediários fazem parte de um conjunto dominante.

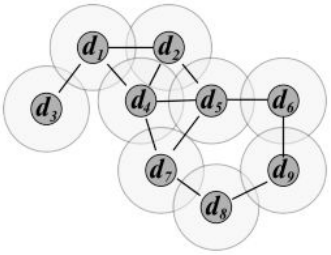

Figura 2: Exemplo de grafo com sensores conectados com base em suas distâncias

4. Algoritmo Exato

Um algoritmo desenvolvido para resolver um problema da forma mais direta, sem nenhuma técnica é chamado de algoritmo exato e realiza sua execução na "força bruta", ou seja, são testadas todas as possibilidades possíveis uma a uma.

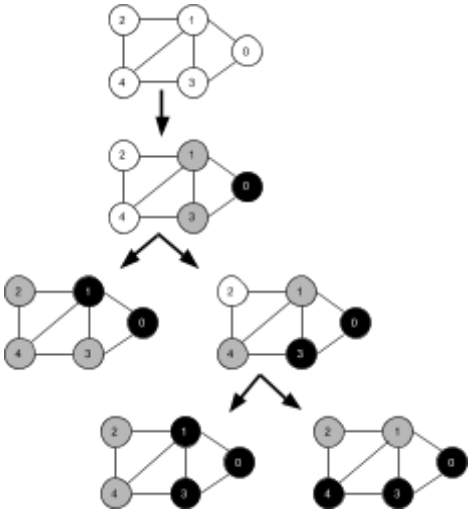

Figura 3: Exemplo do processo usando o algoritmo exato

5. Algoritmo Aproximado

Para a otimização do problema foi aplicado a técnica de busca gulosa sobre o algoritmo, procurando sempre pelo vértice adjacente de maior grau. O algoritmo utilizado foi desenvolvido por Irene Ginani para uma mesma aplicação do caso apresentado anteriormente. O algoritmo guloso utilizado no desenvolvimento foi proposto por David S. Johnson. Como o seu processo consiste em a partir de um vértice acessar um adjacente de grau maior então a complexidade se resume a O(1 + lnΔ) onde Δ é o grau máximo do grafo.

6.1 Pseudocódigo

Algorithm 1 WCA modificado

```
function ALGORITMO WCA(M_nxn, φ, f, v1, v2, v3, v4)
    for i = 0 to n do
        for j = 0 to n do
            if M[i][j] > 0 and M[i][j] < f then
                g ← g + 1                          ▷ g = Grau do vértice
                d ← d + M[i][j]                    ▷ d = Somatório das distancias das arestas
            end if
        end for
```

$dif \leftarrow |grau - \phi|$ ▷ ϕ = Número ideal de vizinhos
Gera um valor para m ▷ m = Mobilidade do nó
Gera um valor para p ▷ p = Potência do nó
$W \leftarrow d * v1 + dif * v2 + m * v3 + p * v4$ ▷ W = Média ponderada
$V[i] \leftarrow v(W)$ ▷ V = Vetor de vertices
end for
$V \leftarrow sort(V)$
while $V \neq \emptyset$ **do**
$v \leftarrow V[0]$
Adiciona as arestas que envolvem o v em MF_{nxn}
Remove o v e seus vertices vizinhos do vetor V
end while
return MF_{nxn} ▷ Nova matriz de adjacencia
end function

6. Testes

Os testes foram realizados em ambiente Linux, no sistema operacional Ubuntu 18.04. O algoritmo foi executado pelo terminal utilizando a configuração **time**. Para o algoritmo exato foram testadas 5 entradas diferentes como visto na imagem abaixo.

Figura 4: Testes do algoritmo exato

Referências

Github - Conjunto Dominante (Irene Ginani)

https://github.com/IreneGinani/Conjuntos-Dominantes/blob/master/relatorio.pdf

Chapter 8 DOMINATING SETS

https://disco.ethz.ch/courses/ss04/mobicomp/lecture/8/Chapter8DominatingSets4Slides.pdf

A Simple Algorithm for Dominating Set

http://webhome.cs.uvic.ca/~wendym/courses/425/14/notes/425_03_dom_alg.pdf