Analise se cobertura mínima de vértice usando para a implementação de número mínimo de semáforo.

Janderson Almeida, Rodrigo Oliveira, Hebert Rocha

Departamento da Ciência da Computação – Universidade Federal de Roraima Pará (UFRR) – Boa Vista – RR– Brasil

jandersonatual62@gmail.com, rodrigoarkhamcity@gmail.com, herbert.rocha@ufrr.com

Abstract. *This article proposes to use the algorithm of minimum coverage of vertices, where it intends to position the minimum of traffic around the city covering as many streets as possible, resulting in cost savings for the city.*

Resumo. *Este artigo propõe a utilização o algoritmo de cobertura mínima de vértices, onde tem com a intenção de posicionar o mínimo de semáforos pela cidade abrangendo o máximo de ruas possíveis, como resultado, trazendo economia de gastos para a prefeitura.*

1. Introdução

Desde antigamente a humanidade recorre a meios de locomoção por motivos de buscar lugares com oportunidades ou transportar alguma coisa, no caso dos nômades que viviam mudando de lugar por meio de animais ou a pé em busca de alimentos, com tempo a necessidade foi aumentando tanto para transporte de pessoas como para transporte de cargas com quantidade muito alta, conforme a necessidade a humanidade foi se aperfeiçoando com maquinas locomotivas como automóveis e trens que transportava pessoas e matérias em distancias muito grande em tão pouco tempo.

Assim, com o surgimento dos primeiros automóveis surgiu também alguns problemas, com o preço acessível e rápida locomoção mais pessoas possuíam seu próprio automóvel, aumentando a quantidade de carros nas ruas onde com consequência a bagunça de trafego dos carros pelas ruas, sem nenhuma regra ou fiscal para orientar surgiu os primeiros acidentes. Mesmo com algumas leis implantadas o fluxo de carros continuavam a aumentar, eis que surge o primeiro semáforo que era para controlar o fluxo e prevenir acidentes que ocorriam muito nos cruzamentos por não ter uma ordem de passagem.

Semáforos em uma cidade é essencial principalmente em uma cidade grande, pois, controla tanto o fluxo e reduz a taxa de acidente, com isso, a prefeitura gasta muito dinheiro para planejar os melhores locais para se implantar um semáforo, tentando abranger o máximo de ruas possíveis.

O problema de encontrar uma cobertura de vértices mínima é um clássico problema de otimização em ciência da computação e é um exemplo típico de um problema de otimização NP-difícil que tem um algoritmo de aproximação. Este trabalho propõem a seguinte pergunta: *Como saber a quantidade mínima de semáforos aproximado que será necessário para ser implantado em uma cidade abrangendo todas as ruas?*

2. **NP-completo**

Informalmente, um problema está na classe NPC – e vamos nos referir a ele como um problema NP-completo – se ele está em NP e é tão "difícil" quanto qualquer problema em NP. (Cormen, 2009)

A classe NP-completo são problemas NP que possui a característica de um que um deles poder ser resolvido em tempo polinomial então todo problema NP-completo terá uma solução em tempo polinomial. (Edson Preste, 2011)

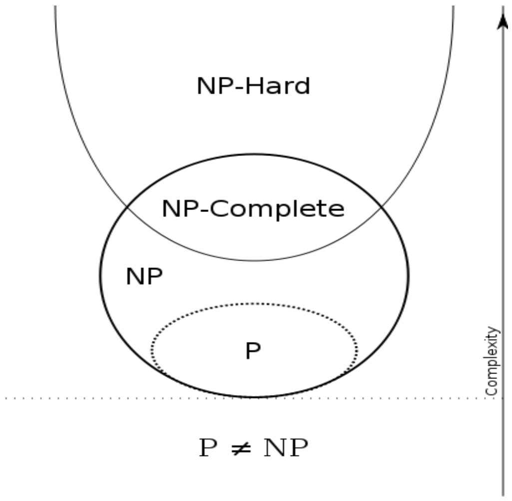

Figura 1. Diagrama de Euler para o conjunto de problemas P, NP, NP-completo e NP-hard.

Fonte: Wikipédia, 2017.

3. **Cobertura mínima de vértice**

Segundo Siqueira (2011), a teoria dos grafos tem uma origem relativamente recente (século XVIII) na história da matemática, sendo seu surgimento estabelecido em 1736, ano da solução do problema das pontes de Königsberg por Euler. Além deste, poucos trabalhos surgiram até meados do século XIX, destacando-se o de Kirchhoff que, em 1847, utilizou modelos em árvores no estudo de circuitos elétricos, e o de Cayley, que utilizou o conceito de grafo para fazer a enumeração dos isômeros dos hidrocarbonetos alifáticos saturados, em química orgânica.

A partir daí diversos problemas surgiram dentro da teoria dos grafos, dentre elas o problema da cobertura mínima de vértices.

Uma cobertura de um grafo é qualquer conjunto de vértices que contenha pelo menos uma das pontas de cada aresta. Em outras palavras, um conjunto X de vértices é uma cobertura se toda aresta do grafo tem pelo menos uma de suas pontas em X. (IME – USP, 2013)

O problema de cobertura de vértices é encontrar uma cobertura de vértices de tamanho mínimo em um dado grafo não orientado. Chamamos tal cobertura de vértices uma cobertura de vértices ótima. Esse problema é a versão de otimização de um problema de decisão NP-completo. Uma cobertura de vértices mínima é uma cobertura de vértices de menor tamanho possível. (Paulo Afonso, 2012)

4. **Heurística**

No algoritmo 1 é descrito o pseudo código da função de pesquisa binaria. Analisando a função é trivial observar que a complexidade obtida de é de $O\,(E * ({}^{V}C_{V/2} + {}^{V}C_{V/4} + {}^{V}C_{V/8} + \ldots \text{ upto } {}^{V}C_{k}))$. Esses termos não são mais do que log (V) no pior dos casos.

Algoritmo 1 – Cobertura mínima de vértices usando busca binaria

```
enquanto (conjunto<limite){
        reinicialize o array para cada subconjunto vértices visitados;
        para v e j de 1 até j < limite; j = j++, v++;{
                se (conjunto & j){
                para (k de 1, até k<=V,  k++){
                        visitado[v][k] = 1;
                        visitado[k][v] = 1;
                        contador++;
                }
        }
        se (contador == número de arestas){
                return true;
        }
        return false;
}
```

No algoritmo 2 é descrito o pseudo código da função de lista de adjacência. A complexidade do tempo do algoritmo acima é O (V + E).

Algoritmo 2 – Cobertura mínima de vértices usando lista de adjacência

```
Para (u de 0, até v, u++){
        se (visitado[u] == falso){
        para (i do começo da lista de adjacências de u até o final da lista de
        adjacências de u, i++){
                int v = *i;
                se (visitado[v] == falso){
                    visitado[v] = true;
                    visitado[u] = true;
                }
        }
}
```

5. Resultados Obtido

Testamos tanto na versão 1 que usa que usa algoritmo aproximada de lista de adjacência como a versão 2 que usa o algoritmo de busca binaria com matriz de adjacência, a tabela 1 mostra os resultados obtidos pelo nosso programa.

Teste	Quantidade de vértices	Quantidade de arestas	Versão 1		Versão 2	
			Resultado	Tempo (µs)	Resultado	Tempo (µs)
data595-1	595	27856	592	1500	595	15657206
data760-1	760	41263	758	2531	760	320055058
data1150-1	1150	80072	1148	4073	1150	504119549
Data1400-1	1400	110038	1398	5005	1400	888862075

Tabela 1.

Referências

Cormen, T. H., Leiserson, C. E., Rivest, R. L., e Stein, C. (2009). Introduction to algorithms. McGraw-Hill Science/Engineering/Math, 3 edition.

Siqueira A.. (2011) “Coloração total equilibrada em subfamílias de grafos regulares”, http://objdig.ufrj.br/60/teses/coppe_d/AngeloSantosSiqueira.pdf, Julho.

IME - USP. (2013) “Uma introdução sucinta a teoria dos grafos”, https://www.ime.usp.br/~pf/teoriadosgrafos/texto/TeoriaDosGrafos.pdf, Julho.

Afonso P.. (2012) “Cobertura de vértices”, http://objdig.ufrj.br/60/teses/coppe_d/AngeloSantosSiqueira.pdf, Julho.