**Documentação do terceiro Trabalho Prático da disciplina de Algoritmos I**

**Luis Gabriel Caetano Diniz – Matrícula: 2019075711**

Departamento de Ciência da Computação - Universidade Federal de Minas Gerais (UFMG)

Belo Horizonte – MG – Brazil

[lgcaetano@ufmg.br](mailto:lgcaetano@ufmg.br)

1. **Descrição do problema**

Neste trabalho, deve-se resolver o seguinte problema: tem-se um conjunto de vilas de uma região agrícola que são representadas por pontos e possuem caminhos que ligam as vilas umas as outras. A administração da saúde pública local deseja que para cada caminho entre duas vilas, pelo menos uma dela possua um depósito de vacinas. O objetivo do trabalho era desenvolver um algoritmo que encontrasse uma solução mínima para conjuntos de vila cujos caminhos não formam ciclos, enquanto na segunda parte deveria se desenvolver uma heurística que produzisse uma solução no máximo duas vezes pior que a solução ótima.

Podemos modelar o problema como um problema de grafos, onde cada vértice representa uma vila e cada aresta um caminho entre duas vilas. Desta forma, precisamos encontrar um conjunto de vértices de forma que qualquer aresta do grafo esteja ligada a pelo um vértice deste conjunto. Este é um problema velho na teoria dos grafos, denominado cobertura de vértices. O que queremos na primeira parte do trabalho é uma solução mínima para o problema da cobertura de vértices.

**2. Estruturas e Algoritmos Utilizados**

**2.1 Estruturas de dados**

Para o desenvolvimento da solução foi necessária a implementação de apenas uma estrutura de dados: Grafos. Foi implementada então a classe Grafo, que possui uma matriz de adjacência para representar as arestas de uma instância da classe. Além de um método construtor, a classe Grafo possui métodos para construir arestas, verificar se aresta existe no grafo e os métodos que são usados em nosso problema: o método getSizeOfMinVertexCover, que obtém nossa solução ótima para a primeira tarefa do problema e vertexCoverEstimate, que implementa a heurística usada para solucionar a segunda parte do trabalho.

**2.2 Algoritmos**

O algoritmo utilizado para encontrar a solução ótima na primeira parte do problema foi o algoritmo de Hopcroft-Karp, que é um algoritmo capaz de encontrar um máximo de cardinalidade de acoplamento em grafos bipartidos. Apesar de aparentemente não possuir nenhuma relação, ao menos inicialmente, com o nosso problema, logo fica claro porque o algoritmo de Hopcroft-Karp é capaz produzir uma solução ótima para o problema da cobertura de vértices em grafos sem ciclos.

Primeiramente, é importante lembrar que, por não possuírem ciclos, os grafos formados pelas vilas na primeira tarefa do trabalho são bipartidos. Isso ocorre porque a única condição para que um grafo seja bipartido é a ausência de ciclos ímpares na sua composição. Contudo, algo ainda soa estranho: o algoritmo mencionado não diz respeito ao problema de cobertura de vértices, certo? Não, pois existe um teorema, denominado teorema de König, que descreve uma equivalência entre o problema de máximo de cardinalidade de acoplamento e o problema da cobertura de vértices mínima em grafos bipartidos. O problema do máximo de cardinalidade de acoplamento consiste em encontrar o maior conjunto de arestas possível em um certo grafo tal que nenhuma aresta compartilhe vértices com outra aresta do mesmo conjunto.

O teorema prova essa equivalência fornecendo uma solução para o problema da cobertura mínima de vértices a partir de uma solução para o problema do máximo de cardinalidade de acoplamento:

Seja um grafo bipartido G(V, E) e sejam L e R as bipartições de V. Suponha que M seja uma solução para o máximo de cardinalidade de acoplamento. Por definição, uma cobertura de vértices em G contém pelo menos um vértice de cada aresta em G e, como M possui arestas que não compartilham vértices, cada vértice contido em uma cobertura de vértices em G pode cobrir apenas uma aresta de M. Portanto, para cada aresta contida em M, uma cobertura de vértices em G possui pelo menos um vértice, ou seja, o número de arestas em M serve como limite inferior para o número de vértices em uma cobertura de vértices em G. Com isso provamos que, se existe uma cobertura de vértices de tamanho |M| em G, ela deve ser mínima.

O teorema vai além e nos dá um algoritmo que nos permite construir uma cobertura de vértices de tamanho |M|, provando que |M| sempre será o tamanho de uma cobertura de vértices mínima em um grafo bipartido: seja U o conjunto de vértices desacoplados em L e Z os vértices que estão em U ou conectados a U por caminhamentos alternado (caminhamentos que alternam entre arestas contidas no acoplamento máximo e arestas que não estão contidas no acoplamento máximo).

Seja K = (L\Z) U (R∩Z), toda aresta do grafo pertence a um caminho alternado (estando conectada a K por um vértice pertencente a R) ou está conectada a K por um vértice pertencente a L. Isso ocorre devido ao fato de que se uma aresta *e* está contida no acoplamento, mas não está em um caminho alternado, então sua ponta que está ligada a um vértice em L não pode estar em um caminho alternado (já que duas arestas do acoplamento não podem compartilhar um vértice entre si) pertencendo então a L\Z. Se *e* não estiver contida no acoplamento e nem em um caminho alternado, então sua ponta que está ligada a um vértice pertencente a L não pode estar contida em um caminho alternado, pois este caminho poderia ser estendido com a adição de *e.* Portanto, K forma uma cobertura de vértices.

Além disso, todo vértice contido em K está conectado a uma aresta contida no acoplamento M, já que L\Z consiste apenas de vértices conectados ao acoplamento, já que Z contém todos os vértices que não estão conectados a uma aresta contida no acoplamento. Toda aresta de R∩Z também deve estar ligada a uma aresta do acoplamento já que se existisse um caminho alternado para um vértice que não está ligado a nenhuma aresta do acoplamento então uma mudança no acoplamento que se resumisse a trocar as arestas acopladas do caminho alternado pelas arestas desacopladas iria aumentar a cardinalidade do acoplamento. Contudo, nenhuma aresta de M pode ter ambas as suas pontas conectadas a um vértice de K. Ou seja, para cada aresta de M, temos um vértice que está em K, nem mais nem menos. Portanto, o número de vértices de K é igual ao número de arestas de M, ou seja |K| = |M|. K é uma cobertura mínima de vértices.

Com isso, provamos que a cardinalidade da solução ótima para o problema do máximo de cardinalidade de acoplamento é igual à cardinalidade da solução ótima para o problema da cobertura mínima de vértices em grafos bipartidos.

Como na primeira tarefa do trabalho é necessário apenas encontrar qual a cardinalidade da solução mínima para o problema de cobertura de vértices, para que resolvamos a primeira tarefa basta que encontremos uma solução para o problema do máximo de cardinalidade de acoplamento e retornemos sua cardinalidade.

É neste momento que aplicamos o algoritmo de Hopcroft-Karp, já que ele nos dá tudo que precisamos.

O algoritmo consiste num loop em que se tentam encontrar *augmenting paths* (APs) no grafo à exaustão, aprimorando o nosso acoplamento a cada iteração. Um AP é um caminho alternado entre dois vértices livres (que não estão conectados a nenhuma aresta do acoplamento). A sua existência implica no acoplamento atual encontrado não ser ótimo, já que sendo M o acoplamento e P, o AP, a diferença simétrica entre M e P formaria um acoplamento de cardinalidade |M| + 1, o que é melhor que M.

Sejam U e V bipartições de um grafo G e seja M o acoplamento que estamos buscando aprimorar. A cada iteração do loop use um algoritmo BFS para particionar o grafo em camadas. Use os vértices livres em U como vértices iniciais desta busca. A busca então deve seguir apenas caminhos alternados em relação a M e deve terminar na primeira camada em que encontrar vértices livres pertencentes a V (no caso da implementação feita no trabalho, quando o algoritmo encontra um vértice de V que está livre, ele aprimora o acoplamento e encerra a iteração do algoritmo). Neste momento, se averiguou que existe um AP entre a raiz da árvore BFS e este vértice livre de V. Com isto, faça com que M seja igual à diferença simétrica entre M (antigo) e o AP encontrado. Repita este processo até à exaustão, quando não forem encontrados mais APs.

Ao final deste processo, teremos um acoplamento ótimo e sua cardinalidade é igual à cardinalidade da solução ótima para o problema da cobertura mínima de vértices e basta que retornemos este valor para que solucionemos a tarefa 1 do trabalho.

Para resolver a segunda tarefa do trabalho, foi utilizada uma heurística que obtém uma solução no máximo duas vezes pior que a solução ótima.

A heurística basicamente consiste em encontrar um acoplamento maximal, um acoplamento (conjunto de arestas de um grafo que não compartilham arestas) composto de arestas tal que todas as arestas do grafo estão contidas no acoplamento ou compartilham um vértice com uma aresta do acoplamento.

Foi utilizado um simples algoritmo guloso, que pode ser encontrado no método vertexCoverEstimate da classe Grafo no código do trabalho. O algoritmo consiste em um loop que não para até que nosso acoplamento se torne maximal. Para verificarmos se nosso acoplamento já atingiu este estado foi criado uma variável que soma o total de arestas do acoplamento com as arestas adjacentes, chamemos o valor representado por esta variável de X. Dentro do loop, verificamos qual aresta vai aproximar mais nosso acoplamento de se tornar maximal, em outras palavras, qual aresta, se adicionada ao acoplamento, vai fazer com X se aproxime mais do número total de arestas. Em cada iteração este processo é repetido até que nosso acoplamento se torne maximal.

Contudo, ao contrário do que ocorre na tarefa1, não basta que retornemos a cardinalidade do nosso acoplamento maximal. Na verdade, a solução será o dobro deste número neste caso, já que geramos a nossa cobertura de vértices de forma diferente do que na primeira parte. Neste caso, para que seja gerado uma cobertura de vértices a partir do nosso acoplamento, basta pegarmos todos os vértices que estão ligados a arestas contidas em nosso acoplamento e agruparmo-los em um conjunto C, representando a nossa cobertura. Isso ocorre devido ao fato de que todas as arestas do grafo estão no acoplamento (estando então ligadas a dois vértices em C) ou então compartilham um vértice com uma aresta do acoplamento (estando ligada a um vértice de C), e, portanto, C é uma cobertura de vértices.

Como é pedido na tarefa2 que se retorne também quais vértices formam a solução para o problema da cobertura de vértices, temos de retornar os vértices em C também.