Universidade Federal do Rio de Janeiro
CPS845 - Tópicos Especiais em Teoria dos Grafos
Professora: Celina M. Herrera de Figueiredo

Relatório de Resumo das aulas
Aula 04

Ary Andrade Neto

Cainã Perreira

Resumo

Neste trabalho, descreveremos as aulas ministradas pela professora Celina de Figueiredo, para a pós-graduação de Engenharia da Computação, do PESC/COPPE. O curso ministrado é baseado na introdução à coloração em grafos para estudantes da área de ciência da computação.

1 Aula04

Nessa aula, foi visto o algoritmo desenvolvido por Whitesides, 1981 [1], para encontrar cortes cliques em $O(n^3)$. Na próxima seção, nós apresentamos o algoritmo adaptado de seu artigo original para o português.

1.1 Tradução do algoritmo para encontrar cortes clique

Inicialização:

Aqui está um método para determinar se G(V) é um clique e para produzir ou um ciclo sem cordas de tamanho de no mínimo quatro ou um corte clique:

(1) Busque por dois vértices não-adjacentes v e w. Se não houver tais vértices, então com certeza G(V) é um clique e não há cortes clique.

(2) Construa um conjunto M da seguinte maneira. Seja N o conjunto de vizinhos de v, digamos, e coloque em M aqueles elementos de N que têm vizinhos na componente W de G(V-N) contendo w.

(3) Busque por um par de vértices não-adjacentes x e y em M. Se M é um clique, o algoritmo pode terminar, uma vez que M é um corte clique de G(V) separando v de w.

(4) Encontre um caminho sem cordas P, cujos vértices terminais sejam x e y e cujos vértices intermediários estão todos em W.

(5) Tome P junto com v para produzir o ciclo desejado.

Algoritmo

(0) Realizar o procedimento de inicialização descrito acima. Esse procedimento produz um ciclo sem cordas com tamanho de no mínimo quatro, se houver um. Caso contrário, ele produz ou um corte clique ou descobre que G(V) é um clique. Se a primeira situação ocorrer, denote o conjunto de vértices por S e avance para o passo (1). À medida que o algoritmo é tratado, mais vértices serão adicionados ao conjunto S, que em geral não irá permanecer um ciclo.

(1) Encontre um componente C de G(V-S) e seja R o conjunto de vértices em S que têm no mínimo um vizinho em C. Uma vez que G(V) é conexo, R não é vazio.

(2) Busque por um par de vértices não-adjacentes x e y em R. Se tal par existir, o algoritmo pode terminar: R deve de fato formar um corte clique de G(V), uma vez que R separa C de S-R em G(V). Note que S-R certamente não

é vazio, já que S não forma um clique e, portanto, contém vértices fora de R.

(3) Encontre um caminho sem cordas P tal que os vértices finais sejam x e y e todos os vértices intermediários pertençam a C.

(4) Adicione os vértices intermediários de P em S.

(5) Repetir as etapas (1) - (4) até que ou um corte clique seja encontrado ou S englobe todos os vértices de V. Se a última situação acontecer, G não possui corte clique.

1.2 Exercício 1

Aplique o algoritmo do artigo para o Exemplo da Figura 13 do texto, desde a inicialização.

Inicialização:

(1) $v = \{a\}$, $w = \{e\}$

(2) $N = \{b, c, g, h\}$

$W =$ ¿$e, f\}$

$M = \{b, c, g, h\}$

(3) $x = \{b\}$, $y = \{h\}$

(4) $P = \{h, f, b\}$

(5) $S = \{a, b, f, h\}$

Algoritmo:

Iteração 1:

(0) $S = \{a, b, f, h\}$

(1) $C = \{g\}$

$R = \{a, b, f, h\}$

(2) $x = \{b\}$, $y = \{h\}$

(3) $P = \{b, g, h\}$

(4) $S = \{a, b, f, g, h\}$

Iteração 2:

(1) $C = \{c, d, e\}$

$R = \{a, b, f, h\}$

(2) $x = \{b\}$, $y = \{h\}$

(3) $P = \{b, c, h\}$

(4) $S = \{a, b, c, f, g, h\}$

Iteração 3:

(1) $C = \{d, e\}$

$R = \{b, c, f\}$

(2) R é corte clique!

1.3 Exercício 2

Justifique a complexidade de tempo O(n^3) para o algoritmo do artigo.

É possível dividir o problema em dois: encontrar o número máximo de iterações e estimar o custo de cada iteração.

Para o primeiro caso temos que no pior caso o algoritmo inicia com $S = C4$ e a cada iteração S cresce em apenas um vértice até ter tamanho V, neste caso temos $n - 4$ iterações na ordem de $O(n)$.

A complexidade dominante em cada iteração vem do passo (1), etapa essa em que o conjunto de vértices R é determinado a partir dos conjuntos S e C. Por definição, R é formado por todos os vértices de S que possuem adjacência com ao menos um vértice de C. Para testar essas adjacências, é necessário avaliar todos os pares de vértices s e c, onde s pertence a S e c, a C. No pior dos casos, se tanto S quanto C forem formados por $n/2$ vértices, onde n é o número de vértices de G, então será necessário $n^2/4$ combinações, o que nos leva a uma complexidade de $O(n^2)$.

Portanto, temos no pior caso o custo de $O(n^2)$ e o número máximo de iterações na ordem de $O(n)$. Assim, a complexidade do algoritmo é $O(n^3)$

Referências

[1] Sue Hays Whitesides. An algorithm for finding clique cut-sets. 1981.