Chapter 3

Lecture 3

Advanced Topics in Graph Algorithms
©Tel-Aviv University
Spring Semester, 1994

Lecturer : Ron Shamir
Scribe : Dekel Tsur
Lecture 3 : May 5

3.1 Perfect Graphs

3.1.1 p-Cririical Graphs

Definition An undirected graph G is called *p-critical* if G is not perfect, but every proper induced subgraph of G is perfect.

Note: If G is p-critical then for all vertices x of G

$$\alpha(G-x) = k(G-x),\ w(G-x) = \chi(G-x)$$

Theorem 3.1 *If G is a p-critical graph on n vertices, then:*

1. $\alpha(G)w(G) = n-1$
2. *for all vertices x of G, $\alpha(G) = k(G-x)$ and $w(G) = \chi(G-x)$*

Proof: By the Perfect Graph theorem, since G is p-critical we have $\alpha(G)w(G) < n$ and $\alpha(G-x)w(G-x) \geq n-1$ for all vertices x. Since α and w are monotone,

$$n > \alpha(G)w(G) \geq \alpha(G-x)w(G-x) \geq n-1$$

Hence, $\alpha(G)w(G) = n-1$, $\alpha(G) = \alpha(G-x) = k(G-x)$ and $w(G) = w(G-x) = \chi(G-x)$
■

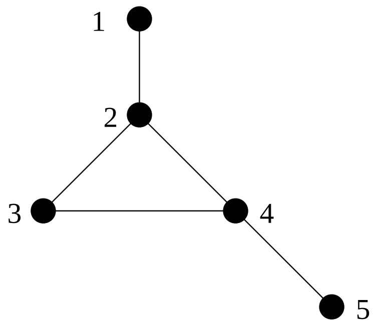
3.1.2 A Polyhedral Characterization of Perfect Graphs

Let A be an $m \times n$ matrix. We define the two polyhedra

$$P(A) = \{\mathbf{x} | A\mathbf{x} \leq \mathbf{1},\ \mathbf{x} \geq \mathbf{0}\}$$

$$P_I(A) = convex - hull(\{\mathbf{x} | \mathbf{x} \in P(A),\ \mathbf{x}\, integral\})$$

where $\mathbf{x}$ is an n-vector, $\mathbf{1}$ is the m-vector of all ones, and $\mathbf{0}$ is the n-vector of all zeros. Clearly $P_I(A) \subseteq P(A)$, and for $(0,1)$ matrix A without zero columns, $P(A)$ and $P_I(A)$ are bounded in the unit hypercube in R^n. The *clique matrix* of a graph $G = (V, E)$ is a $(0,1)$ matrix where if $V = \{1, 2, \ldots n\}$ and $C_1, C_2, \ldots C_m$ are all the maximal cliques of G numbered arbitrarily, then $A_{ij} = 1$ iff $j \in C_i$. See Figure 3.1 for an example.



$$\begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Figure 3.1: A graph and its clique matrix

Theorem 3.2 (Chvátal, 75) *Let A be the clique matrix of an undirected graph G. Then G is perfect iff $P(A) = P_I(A)$.*

Polyhedron P is called *integral* if all the coordinates of all its extreme points are integers, i.e. if $P(A) = P_I(A)$. In other words, theorem 3.2 is: G is perfect iff $P(A)$ is integral polyhedron. In particular, this implies that optimization problems which can be posed as *integer* programming problems with constraints $P_I(A)$, can be solved in polynomial time on perfect graphs, by solving the corresponding *linear* program.

3.1.3 The Strong Perfect Graph Conjecture

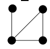
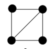
It is easy to see that the odd cycle C_{2k+1} (for $k \geq 2$) is a p-critical graph. By the Perfect Graph theorem its complement $\overline{C_{2k+1}}$ is also p-critical. C_{2k+1} and $\overline{C_{2k+1}}$ (called *odd hole* and *odd anti-hole* respectively) are the only known p-critical graphs. In 1960 C. Berge conjectured that there are no other p-critical graphs. This conjecture is called the *strong perfect graph conjecture* (SPGC).

The strong perfect graph conjecture can be rephrased in several equivalent forms:

(1) G is perfect iff it does not contain induced subgraph isomorphic to an odd hole or an odd anti-hole.

(2) G is perfect iff every cycle of length ≥ 5 in G or $\overline{G}$ contains a chord.

(3) The only p-critical graphs are C_{2k+1} and $\overline{C_{2k+1}}$ (for $k \geq 2$).

The strong perfect graph conjecture remains an open question. However, the conjecture was proved on several classes of graphs like planar graphs, circular arc graphs, graphs with $w \leq 3$, graphs with maximal degree ≤ 6, and also $K_{1,3}$ - free, K_4 - free,  -free, and  -free graphs. (a graph is *X-free* if it does not contains induced subgraph isomorphic to X).

3.2 Triangulated Graphs

3.2.1 Introduction

Definition An undirected graph G is called *triangulated* if every cycle of length > 3 contains a chord. In other words G is triangulated if it does not contain an induced subgraph isomorphic to C_n for $n > 3$.

Being triangulated is a hereditary property, meaning that every induced subgraph of a triangulated graph is also triangulated. Triangulated graphs have also been called *chordal*, *rigid-circuit*, *monotone transitive* and *perfect elimination* graphs. For an example of triangulated graph see Figure 3.2.

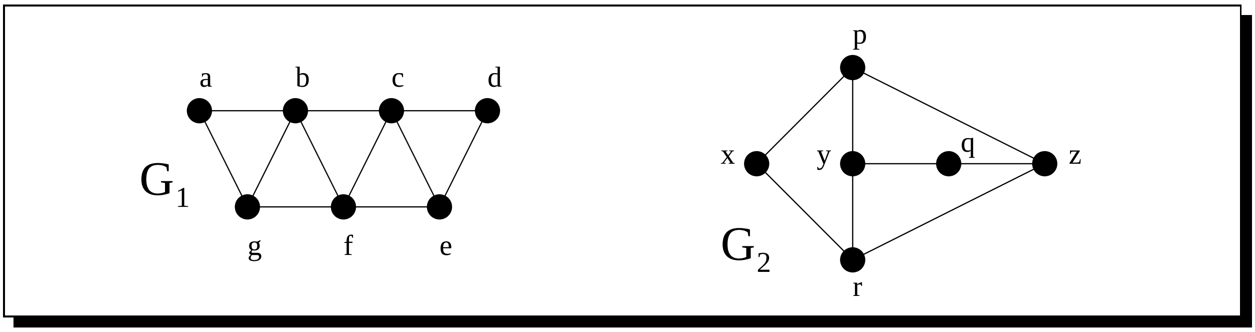


Figure 3.2: Two graphs, the left one is triangulated, and the right one is not.

3.2.2 Characterizing Triangulated Graphs

Definition A vertex x in G is called *simplicial* if $Adj(x)$ induces complete graph in G (i.e.,

$Adj(x)$ is a clique).

We will prove that every triangulated graph has a simplicial vertex. Since being triangulated is a hereditary property, it gives us an iterative procedure for identifying triangulated graphs: Search for simplicial vertex, remove it from the graph and repeat until there is no simplicial vertex (in that case, the graph is not triangulated) or there are no vertices left (and then the graph is triangulated).

Definition Let G be an undirected graph and let $\sigma = [v_1, \dots, v_n]$ be an ordering of the vertices. σ is called *perfect elimination order* (PEO) if every v_i is a simplicial vertex in the induced subgraph $G_{\{v_i,\dots,v_n\}}$. In other words, each set $X_i = \{v_j \in Adj(v_i) | j > i\}$ is complete.

In Figure 3.2, $\sigma = [a, g, b, f, c, e, d]$ is one of the perfect elimination orders of G_1. In contrast to this, G_2 has no simplicial vertex, so obviously it has no perfect elimination order.

Definition A subset $S \subset V$ is called a *vertex separator* for nonadjacent vertices a, b (or *a-b separator*), if in G_{V-S} a and b are in different connected components. If no proper subset of S is an a-b separator, S is called *minimal vertex separator.*

The example will be again on the graphs in Figure 3.2. In G_2, the set $\{y, z\}$ is a minimal vertex separator for p and q, whereas $\{x, y, z\}$ is a minimal vertex separator for p and r. In G_1 all the minimal vertex separators have 2 vertices which are neighbors.

Theorem 3.3 *(Dirac 61, Fulkerson and Gross 65)*
For undirected graph G the following statements are equivalent:

(1) G is triangulated.

(2) G has a perfect elimination order. Moreover, any *simplicial vertex can start a perfect order.*

(3) Every minimal vertex separator induces a complete subgraph of G.

Proof:

(3) $\to$ (2) Let $[a, x, b, y_1, \dots, y_k, a]$ $(k \geq 1)$ be a simple cycle. Any minimal a-b separator must contain vertices x and y_i for some i, but from (3) it follows that $xy_i \in E$, which is a chord of the cycle.

(1) $\to$ (3) Let S be an a-b separator. We will denote G_A, G_B the connected components of G_{V-S} containing a and b respectively. Since S is minimal, every vertex $x \in S$ is a neighbor of a vertex in A and a vertex in B. For any $x, y \in S$ there exist minimal paths $[x, a_1, \dots, a_r, y] (a_i \in A)$ and $[x, b_1, \dots, b_k, y] (b_i \in B)$. Since $[x, a_1, \dots, a_r, y, b_1, \dots, b_k, x]$

is a simple cycle of length ≥ 4, it contains a chord. For every i, j $a_i b_j \notin E$ (S is a-b separator) and also $a_i a_j \notin E, b_i b_j \notin E$ (by the minimality of the paths) Necessarily $xy \in E$ is the chord.

■

To prove the rest of the theorem, we will first prove the following lemma:

Lemma 3.4 *(Dirac 61) Every triangulated graph G has a simplicial vertex. If G is not complete then it has two nonadjacent simplicial vertices.*

Proof: If G is complete the lemma is trivial. We will prove the lemma for noncomplete graphs by induction on the number of vertices in G. The lemma is trivial for graphs with 1 or 2 vertices. Assume the statement is correct for graphs with $n-1$ vertices and let G be a triangulated graph with n vertices. Let a, b be two nonadjacent vertices in G, and let S be minimal a-b separator. Denote by G_A, G_B the connected components of G_{V-S} which contain a, b respectively. We shall look at G_{A+S}:

1. If G_{A+S} is complete, then each of its vertices is a simplicial vertex in G_{A+S}. Particularly, every vertex in A which is simplicial in G_{A+S} is also simplicial in G (as there are no edges between A and B).
2. If G_{A+S} is not complete, by induction it contains two nonadjacent simplicial vertices. Since S is complete (we already proved $1 \to 3$ in Theorem 3.3) at least one of the simplicial vertices is in A and like before this vertex is simplicial in G.

We showed that A contains a simplicial vertex in G. In the same manner, B contains a simplicial vertex in G. These two vertices are not adjacent. ■

Now we will continue the proof of Theorem 3.3:

(1) $\to$ (2) By Lemma 3.4, G contains a simplicial vertex x. $G_{V-\{x\}}$ is triangulated and contains less vertices, so by induction it has a perfect elimination order $[\sigma_{i_1}, \ldots, \sigma_{i_{n-1}}]$. Since x is simplicial $[x, \sigma_{i_1}, \ldots, \sigma_{i_{n-1}}]$ is a perfect elimination order in G.

(2) $\to$ (1) Let C be a simple cycle in G. Let x be the *first* vertex from C to appear in the perfect elimination order. In the induced graph before eliminating x, the two neighbors of x in the cycle still exist, and since x is simplicial in that graph, these two vertices form a chord in the cycle. Hence, every cycle in G contains a chord.

■

```
assign label 0 to each vertex
for i ← n to 1 by −1 do
    select: pick an unnumbered vertex v with largest label
    σ(i) ← v
    update: for each unnumbered vertex w ∈ Adj(v) do
        add 1 to label(w) end for
end for
```

Figure 3.3: Code for MCS algorithm

3.2.3 Recognizing Triangulated Graphs

If a graph G is triangulated and is not complete, it has two nonadjacent simplicial vertices. Also every induced subgraph of G is also triangulated. Therefore it is possible to choose *any* vertex v and decide it will be last in the PEO (it is always possible to choose another simplicial vertex instead of v). Afterwards, it is possible to choose a vertex u adjacent to v and put it in $(n-1)$st position. It is possible to continue choosing vertices *backwards* to the perfect elimination order. Using the above idea will give us a linear time algorithm for recognizing triangulated graphs. The following linear time algorithm (called Maximum Cardinality Search) by Tarjan and Yannakakis (1985), receives a graph G given by adjacency list for each vertex, and outputs an order σ of the vertices, with the property that if G is triangulated then σ is a PEO. We will show later how to check if this vertex order is a perfect elimination order (i.e. if the graph is triangulated) in linear time. The code is in Figure 3.3, and a running example is given in Figure 3.4.

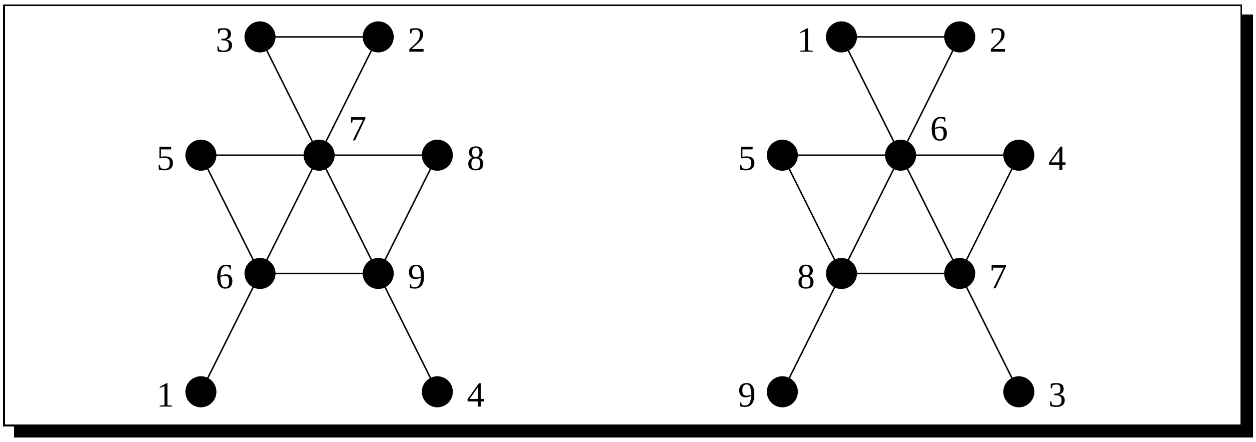


Figure 3.4: Two different possible running of the MCS algorithms.

Time Complexity

Let S_i be the set of all vertices v that have not been numbered yet, and have label$(v) = i$. Each S_i is represented by a doubly linked list. The set of active S_i-s (all those S_i-s which were non-empty at some point) will also be represented by a doubly linked list in increasing order of i,with a pointer to the non-empty set with largest label. For each vertex we will store its label i and a pointer to its position in S_i. When v receives a number we take v out of its list and move each neighbor of v one list upwards. This takes $O(1 + degree(v))$, therefore the overall time complexity is $O(|V| + |E|)$.