ECE 360C — Algorithms **Spring 2007**

Graph Matching

Adnan Aziz *UT Austin*

Based on notes I took from Raimund Seidel's Algorithms class in Spring 1991—some of this is also covered in CLRS, specifically 26.3 (see also Problem 26-7).

Given an undirected graph $G = (V, E)$, a matching M is a subset of E such that every vertex $v \in V$ is incident to at most one edge in M.

Some notation:

- A matching M is *maximal* if $\not\exists \text{matching} M'$ such that $M \subset M'$.
- A matching M is *maximum* if $\not\exists \text{matching} M'$ such that $| M | \leq | M' |$.
- A matching M is *perfect* if each vertex is incident to an edge of M.

Restrict our attention to *bipartite graphs*, i.e., $G = (U \cup V, E)$ where $U \cap V = \emptyset$ and all edges are of the form $\{u, v\}, u \in U, v \in V$.

- Pairing off dance partners
- Pairing repairmen to repairs

How do we find maximum matchings?

Focus on alternating paths.

Given M, an *alternating path* for M is a path from $u \in U$ to $v \in V$, both unmatched, such that the edges on the path alternate between M and $E \setminus M$.

Theorem 1 *A matching M is maximum $\Leftrightarrow$ there is no alternating path for M.*

Proof: ⇒ If there is an alternating path, the matching can be increased in size, so it's not maximum,

⇐ Need to show if the matching M is not maximum then there exists an alternating path for M.

Let P be a maximum matching for G.

Consider the graph G' induced on $P \cup M$ (note that P and M are not necessarily disjoint).

- We'll refer to edges as being pink if they are from P and maroon if they are from M (could be both.

In G' every vertex is incident to at most 1 pink edge and 1 maroon edge. Therefore each vertex is incident to at most two edges. Consequently, connected components of G' are singletons, paths, or cycles.

Specifically, there are only 8 possibilities for the connected components of G', shown in Figure 1.

Some observations:

- Components of type C5 and C7 are impossible (since otherwise P would not be maximum).
- The number of pink and maroon edges in C1, C3, C4, C6, and C8 are the same.

Hence the assumption that M is not maximum means that we must have a component of the form C3, i.e., there's an alternating path. ■

Upshot—to find the largest possible matching, look at alternating paths.

How to systematically find augmenting paths?

Start with a maximal matching M, and create a new directed graph on $V \cup U$, where edges in M go from V to U and edges in $E \setminus M$ go from U to V.

Now do a DFS starting at an unmatched vertex in V—if we find an unmatched vertex, we've found an alternating path.

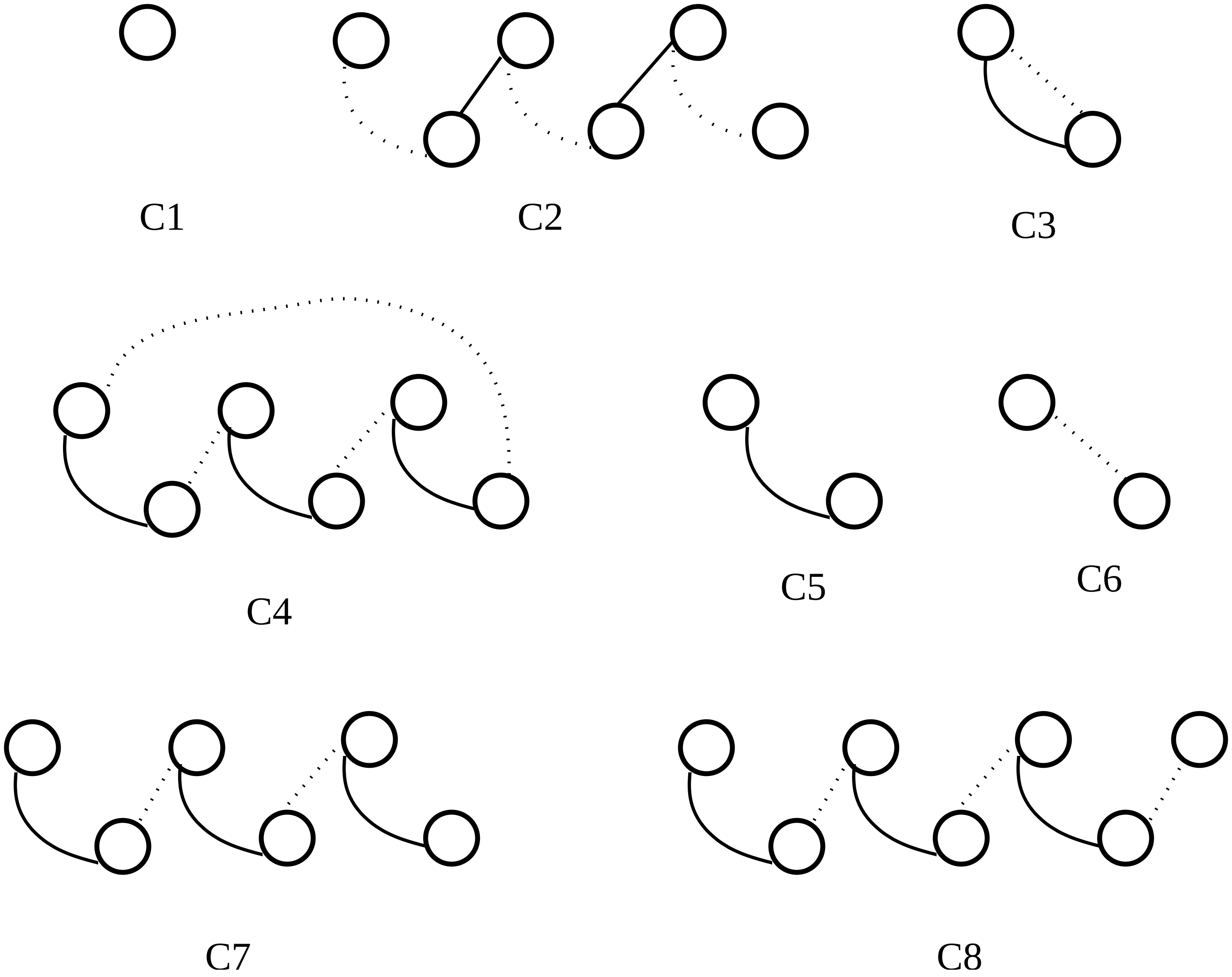

Figure 1: SCCS of G'

Time complexity? n vertices and m edges: each DFS is $O(n + m)$ and there are at most n calls to DFS (since we're reducing the number of unmatched vertices by 2 each time $\Rightarrow O(n(n + m))$.

Better approach Hopcroft/Karp. Ides—use BFS, stop when you hit a layer of unmatched vertices. Can find a collection of alternating paths from this BFS tree. Flip and iterate. Time complexity is $O(\sqrt{n}(n + m))$.

The above was for bipartite graphs. For a long time it was not known for general graphs if there was an efficient algorithm; the only approaches were exponential. Edmonds discovered a $O(n^5)$, and later Micali and Vazirani improved the time bound to $O\sqrt{n}(n + m))$.