Matching in General Graphs
Edmonds' Blossom Algorithm

Dr. Rajiv Misra
Associate Professor
Dept. of Computer Science & Engg.
Indian Institute of Technology Patna
rajivm@iitp.ac.in

Preface

Recap of Previous Lecture:

In previous lecture, we have discussed Matchings in General Graphs, Tutte's 1-Factor Theorem and *f*-Factor of Graphs

Content of this Lecture:

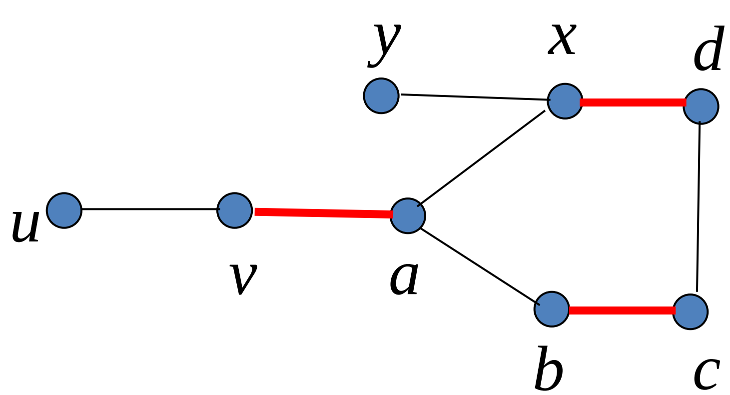
In this lecture, we will discuss the Matchings in General Graphs *i.e.* Edmonds' Blossom Algorithm.

Edmonds' Blossom Algorithm : Matchings in general graphs vs bipartite graphs

- **Berge's Theorem** stated that a matching M in G has maximum size if and only if G has no M-augmenting path.
- We can thus find a maximum matching using successive augmenting path. Since we augment at most $n/2$ times, we obtain a good algorithm if the search for an augmenting path does not take too long. **Edmonds [1965a]** presented the first such algorithm in his famous paper **"Paths, trees, and flowers'.**
- In bipartite graphs, we can search quickly for augmenting paths because we explore from each vertex at most once. An M-alternating path from u can reach a vertex x in the same partite set as u only along a saturated edge. Hence only once can we search and explore x.
- This property fails in graphs with odd cycles, because M-alternating paths from an unsaturated vertex may reach x both along saturated and along unsaturated edges.

Example 3.3.14

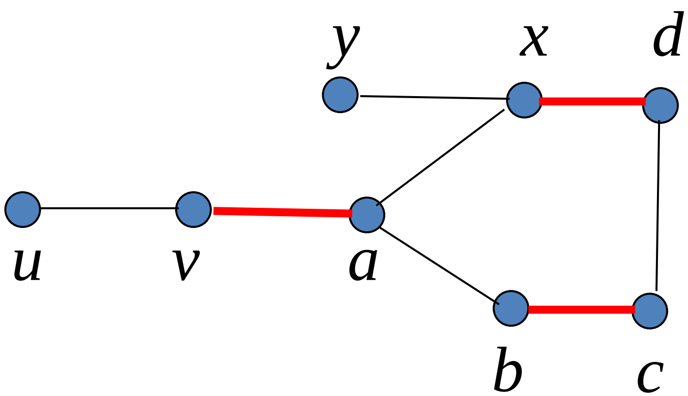
- In the graph below, with M indicated in bold, a search for shortest M-augmenting paths from u reaches x via the unsaturated edge ax. If we do not also consider a longer path reaching x via a saturated edge, then we miss the augmenting path u, v, a, b, c, d, x, y.

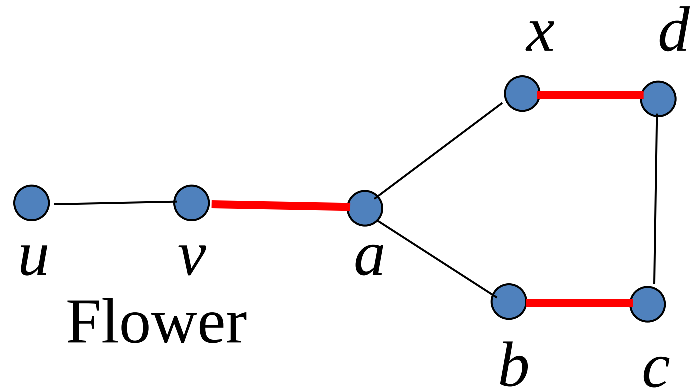


Definitions: Flower, Stem, Blossom

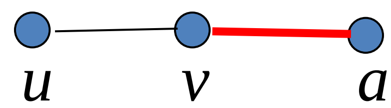
- Let M be a matching in a graph G, and let u be an M-unsaturated vertex.
  - A **flower** is the union of two M-alternating paths from *u* that reach a vertex *x* on steps of opposite parity (having not done so earlier).
  - The **stem** of the flower is the maximal common initial path (of nonnegative even length).
  - The **blossom** of the flower is the odd cycle obtained by deleting the stem.

Example of Flower, Stem, and Blossom

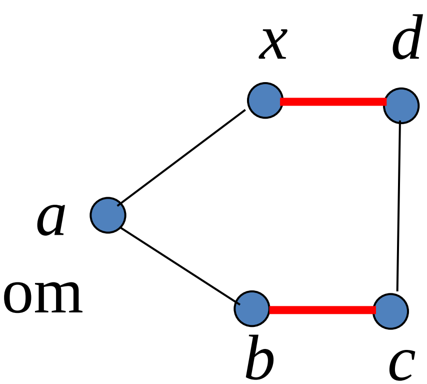




Flower



Stem



Blossom

Algorithm: Edmonds' Blossom Algorithm [1965a]

Input: A graph G, a matching M in G, an M-unsaturated vertex u.

Idea: Explore M-alternating paths from u, recording for each vertex the vertex from which it was reached, and contracting blossoms when found. Maintain sets S and T analogous to those in Algorithm 3.2.1, with S consisting of u and the vertices reached along saturated edges. Reaching an unsaturated vertex yields an augmentation.

Initialization: $S=\{u\}$and $T=\varnothing$

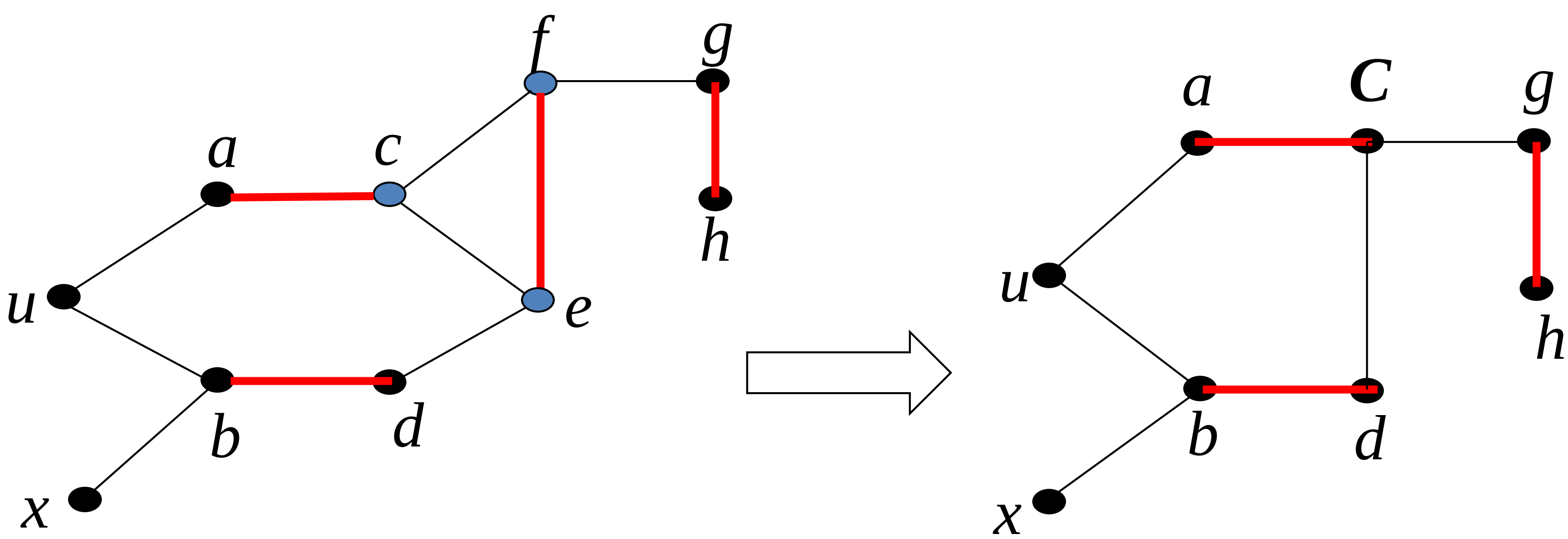
Edmonds' Blossom Algorithm$_{3.3.17}$

Iteration:

- If S has no unmarked vertex, stop; there is no M-augmenting path from u. Otherwise, select an unmarked $v \in S$. To explore from v, successively consider each $y \in N(v)$ such that $y \notin T$.
  - If y is unsaturated by M, then trace back from y (expanding blossoms as needed) to report an M-augmenting u, y-path.
  - If $y \in S$, then a blossom has been found. Suspend the exploration of v and contract the blossom, replacing its vertices in S and T by a single new vertex in S. Continue the search from this vertex in the smaller graph. Otherwise, y is matched to some w by M. Include y in T (reached from v), and include w in S (reached from y).
  - After exploring all such neighbors of v, mark v and iterate.

Example 3.3.16

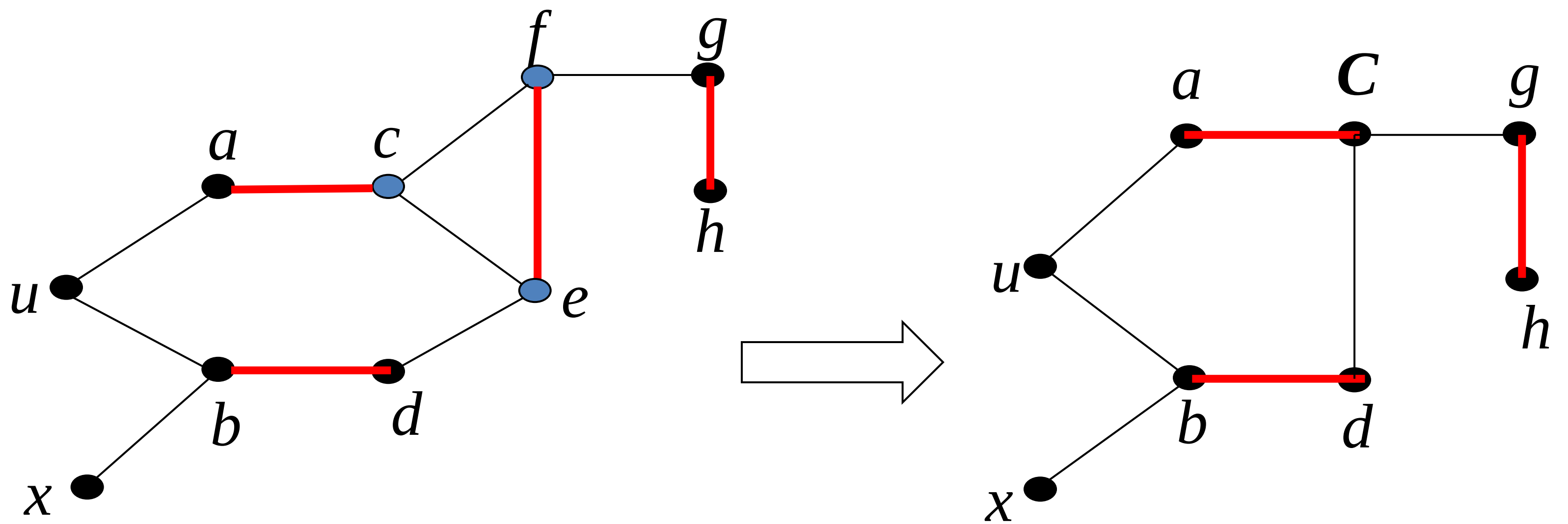
- Let M be the bold matching in the graph on the left below. We search from the unsaturated vertex u for an M-augmenting path. We first explore the unsaturated edges incident to u, reaching a and b. Since a and b are saturated, we immediately extend the paths along the edges ac and bd. Now S={u, c, d}. If we next explore from c, then we find its neighbors e and f along unsaturated edges. Since $e\ f \in M$, we discover the blossom with vertex set {c, e, f}. We contract the blossom to obtain the new vertex C, changing S to {u, C, d}. This yields the graph on the rights.



Example continue

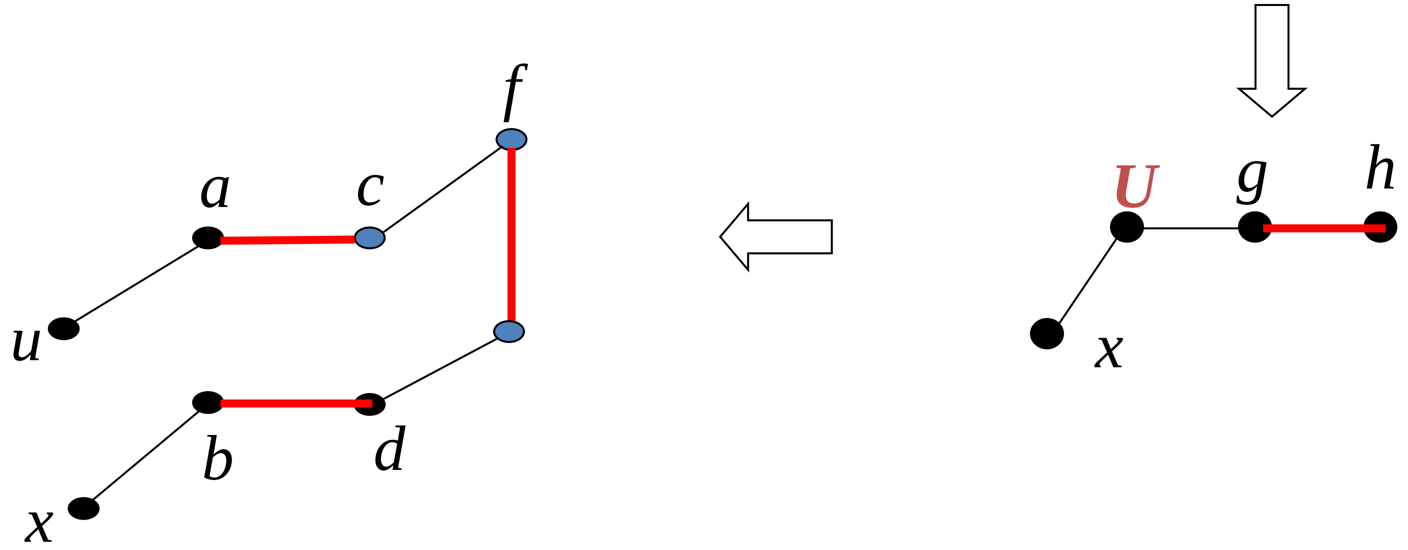
- Suppose we now explore from the vertex *C* ϵ S. Unsaturated edges take us to g and to d. Since g is saturated by the edge *gh*, we place *h* in *S so S={u, C, d,h}*. Since *d* is already in *S*, we have found another blossom. The paths reaching *d* are *u, b, d* and

  u, a, C, d.



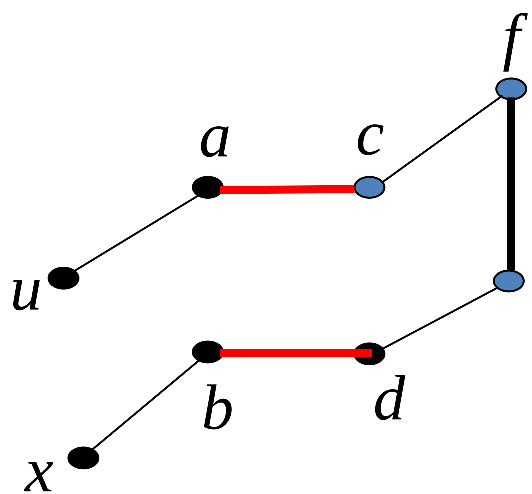
Example continue

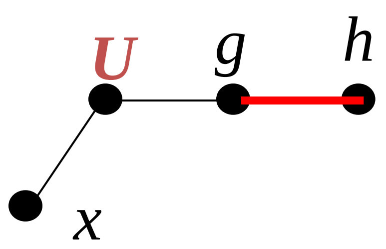
- We contract the blossom, obtaining the new vertex U and the graph on the right below, with $S = \{U, h\}$. We next explore from h, finding nothing new (if we exhaust S without reaching an unsaturated vertex, then there is no M-augmenting path from u). Finally, we explore from U, reaching the unsaturated vertex x.



Example continue

- Having recorded the edge on which we reached each vertex, we can extract an *M*-augmenting *u*, *x*-path. We reached *x* from *U*, so we expand the blossom back into {*u*, *a*, *C*, *d*, *b*} and find that *x* is reached from *U* along *bx*. The path in the blossom *U* that reaches *b* on a saturated edge ends with *C*, *d*, *b*. Since *C* is a blossom in the original graph, we expand *C* back into {*c*, *f*, *e*}. Note that *d* is reached from *C* by the unsaturated edge *ed*. The path from the "base" of *C* that reaches *e* along a saturated edge is *c, f, e*. Finally, *c* was reached from a and a from u, so we obtain the full augmenting path ***u, a, c, f, e, d, b, x*****.**





Remark 3.3.18

- **Edmonds' original algorithm runs in time O(n^4).** The implementation in Ahuja-Magnanti-Orlin [1993] runs in time O(n^3). This requires: (1) appropriate data structures to represent the blossoms and to process contractions, and (2) careful analysis of the number of contractions that can be performed, the time spent exploring edges, and the time spent contracting and expanding blossoms.
- The first algorithm solving the maximum matching problem in less than cubic time was the O($n^{5/2}$) algorithm in Even-Kariv [1975]. The best algorithm now known runs in time O($n^{1/2}m$) for a graph with n vertices and m edges (this is faster than O($n^{5/2}$) for sparse graphs). The algorithm is rather complicated and appears in Micali-Vazirani [1980], with a complete proof in Vazirani [1994]

Conclusion

- In this lecture, we have discussed the Edmonds' Blossom Algorithm and also discuss the concepts of flower, stem and blossom.

- In upcoming lectures, we will discuss the Connectivity and Paths.