CS 6200: Algorithmics II	**Fall 2020**
Instructor: Avah Banerjee, S&T.	
Lectures: 16	**Dates: 9/30**

1 Matching

Theorem 1.1 *Suppose G' be the graph formed after shrinking a blossom b. Then G' contains an augmenting path if and only if G does.*

Proof: Suppose G' contains an augmenting path, say p.
Case-1: As shown in Figure 1, the path p does not go through the blossom. Then augmenting path p in the grapth G' is also augmenting path in the graph G.

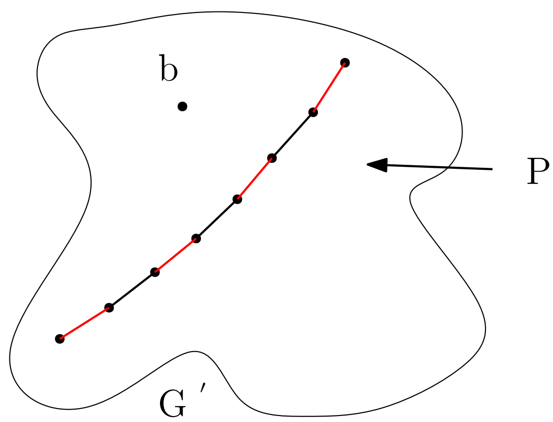

Fig. 1: The graph G in theorem 1.1.

Case-2: As shown in Figure 2, if the base of the blossom is not free, change the matching in G by switching the edges along the stem to make the base free (see Figure 2 below). Then the blossom has a free vertex as a base. Note that the switching does not change the matching size in either graph. In this way, we need consider only the case in which the base of the blossom is free.

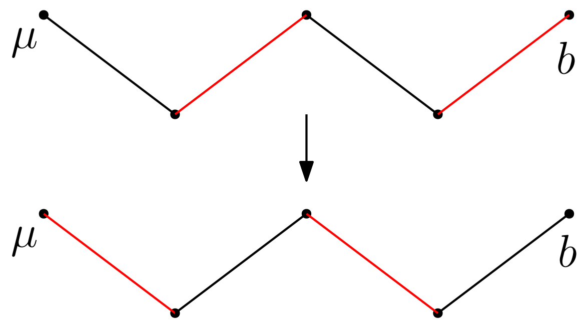

Fig. 2: The edge switching graph to make the base free.

As shown in Figure 3, suppose G' has an augmenting path. Either this path is an augmenting path in G, or it ends at the blossom, and it can be extended to an augmenting path in G by following the blossom in the direction that results in alternation until reaching the base.

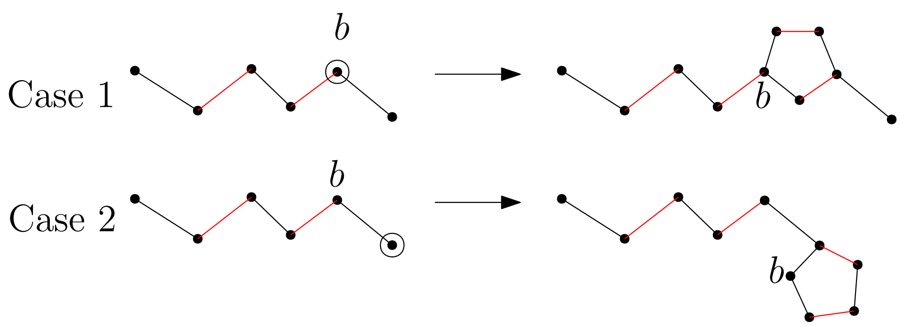

Fig. 3: Different cases in the proof.

Note that the b might not always be the center of the blossom. In Figure 4, suppose there exists a graph in Figure 4 (a), and the Figure 4 (b) was obtained when b was shrunk. Suppose an augmenting path was found from u, and there is always an augmenting path until they reach the b. The b here was the edge going to be the last edge because b was a free vertex, but there would be a free edge as shown in Figure 4 (c), where b is not the center of the blossom. ■

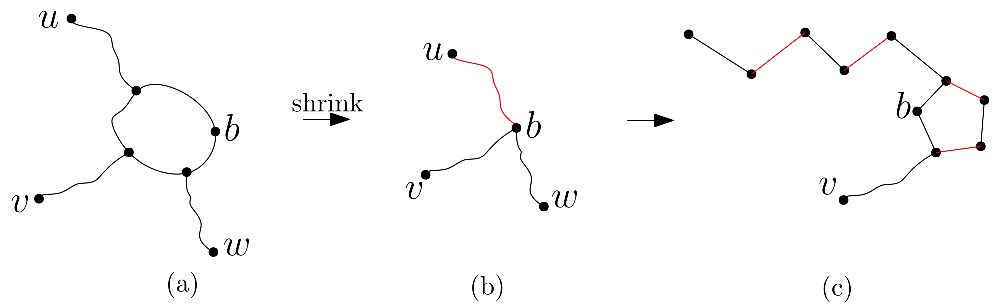

Fig. 4: Cases where b is not the center of the blossom.

If there exist more than one blossoms in a graph, the argument in the proof above can be applied to find augmenting paths when expanding the shrunk blossoms one by one. This process is shown in figure 5 (a)-(c).

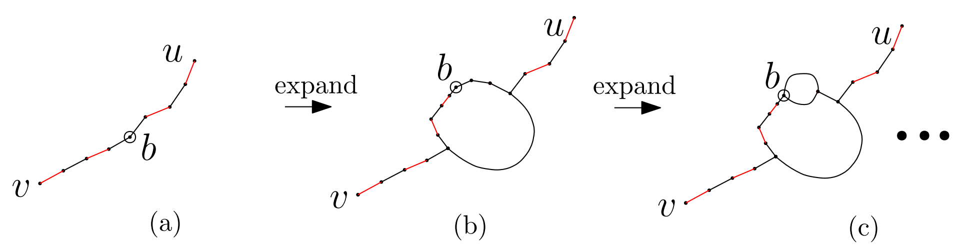

Fig. 5: Cases where more than one blossoms exist.

The Blossom Shrinking Algorithm is shown below in Algorithm 1.

Algorithm 1 Blossom Shrinking Algorithm

Input: a graph G , and a matching M;
Output: an augmenting path or a blossom;

Initialization: All vertices are unlabeled.
while $\exists$ an unlabeled free vertex μ. Label it [μ, even]
    or $\exists$ an unexamined edge μ, ν with μ labeled [ω, even] **do**
    if ν is unlabeled and free **then**
        augmenting path found.
    else if ν is unlabeled, but matched to x **then**
        label (ν)=[ω, odd]
        label (x)= [ω, even].
    else if label(ν)=(y, even) and $\omega \neq y$ **then**
        augmenting path found /*μ and ν are in separate trees*/
    else if label(ν)=(ω, even) **then**
        blossom found
        Let $r = lca(\mu, \nu)$,
        for the tree roofed at ω **do**
            make r the base of the blossom
        end for
    else
        do nothing
    end if
end while

Correctness: The algorithm finds an augmenting path if and only if G has one augmenting path.

Proof: Suppose p is an augmenting path in Figure. 6, since μ, ν are free, algorithm will label both μ, ν as even. For each matched edged either both ends are unlabeled, or only one is even and the other is odd. Therefore, there exists a free edge whose ends are both even. This edge will be examined by the algorithm before the algorithm stops, such that a blossom or an augmenting path will be found. Note that this works for both cases in Figure. 6.

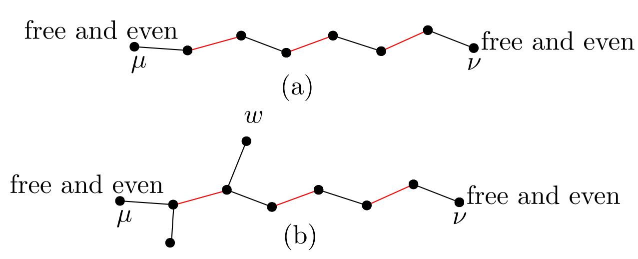

Fig. 6: Different cases in the proof of the correctness.

■