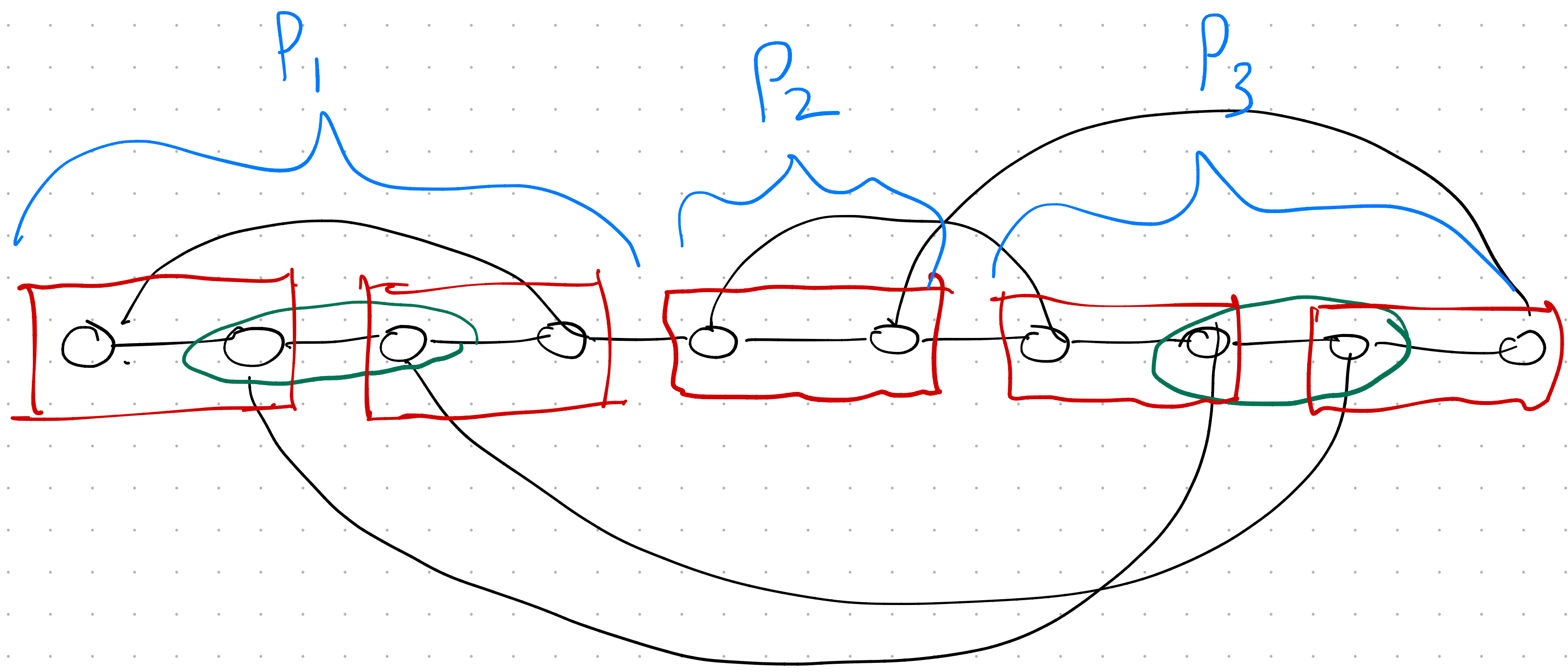
28 Aug 2024

Algorithms for Max Bipartite Matching



P_1, P_2, P_3 are all M-augmenting paths and they are vertex-disjoint. (No vertex belongs to more than one of the paths.)

Lemma. If G is a graph and M is a matching G then the following are equivalent for all $k \in \mathbb{N}$

(i) there exists a matching M' with $|M| + k$ edges

(ii) G contains k vertex-disjoint M-augmenting paths

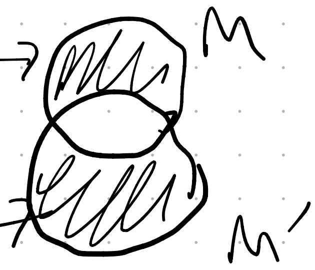
(iii) there exists matching M' s.t. the edge set $M \oplus M'$ contains k vertex-disjoint M-augmenting paths.

Proof: (i) $\Rightarrow$ (iii) Consider M' such that $|M'| = |M| + k$.

$M \oplus M'$ contains 2 types of edges.

"green" – in M, not M'

"red" – in M', not M.



Exactly k more edges of the 2nd type than the 1st.

$(V(G), M \oplus M')$ is a graph of max-degree 2, a union of vertex-disjoint paths and cycles alternating w.r.t. M.

- cycles are of even length and have equal # of red as green edges.

- paths are of unconstraint and # red = # green + $\{-1, 0, 1\}$.

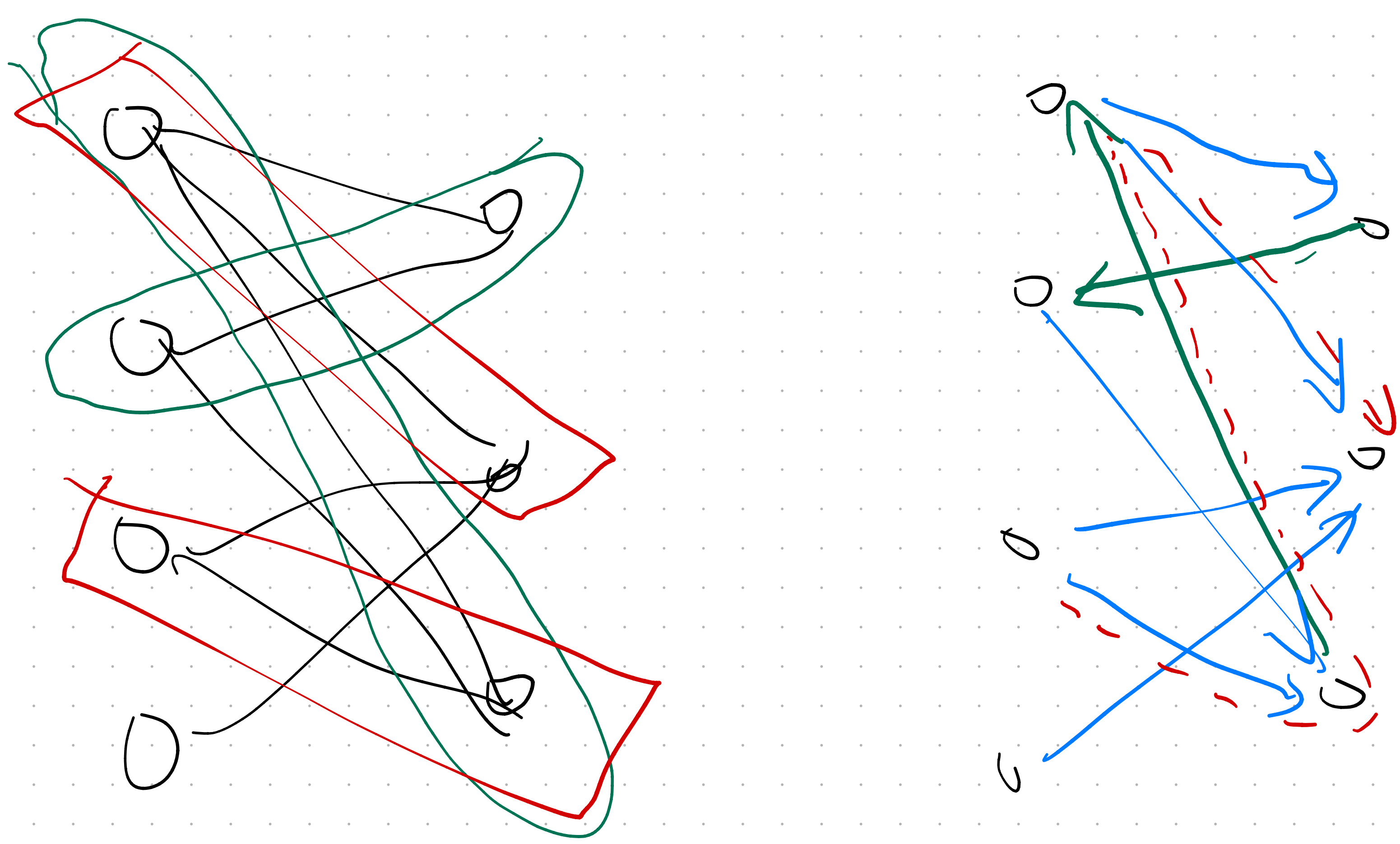
At least k paths have # red = # green + 1. These are the k vertex-disjoint M-augmenting paths asserted in (iii).

(iii) $\Rightarrow$ (ii) immediate

(ii) $\Rightarrow$ (i) check that $M \oplus$ (union of k disj. aug paths) is a matching s.t. $|M'| = |M| + k$.

Assume G bipartite:

$$V(G) = L \cup R \qquad L, R \text{ disjoint.}$$
$$E(G) \subseteq L \times R$$



$$G \Longrightarrow G_M$$

If $F = \{\text{free vertices}\}$ then

M-augmenting paths in G are in 1:1 corresp. with paths in G_M from $L \cap F$ to $R \cap F$.

BFS takes $O(|E| + |V|) = O(m + n)$

Overall running time $= O(mn + n^2)$.
(Improves to $O(mn + n)$.)

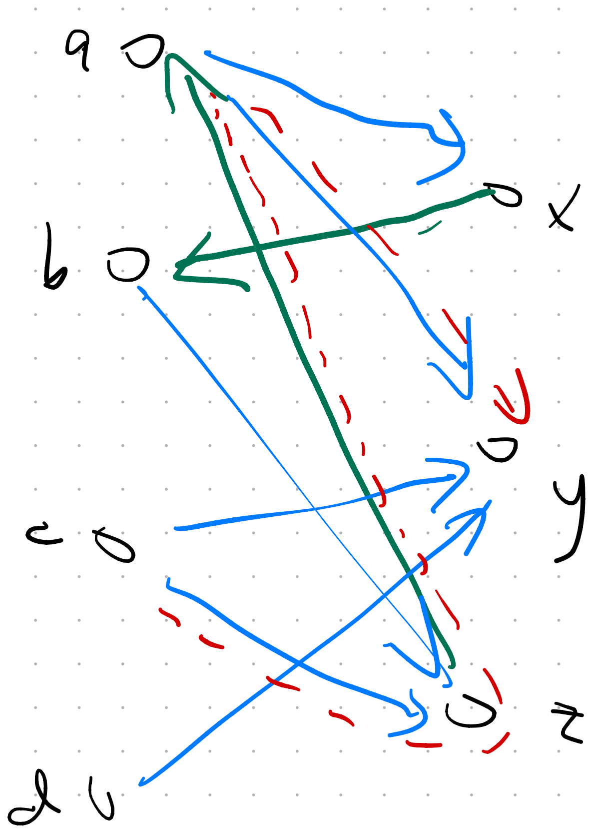
Hopcroft–Karp Algorithm

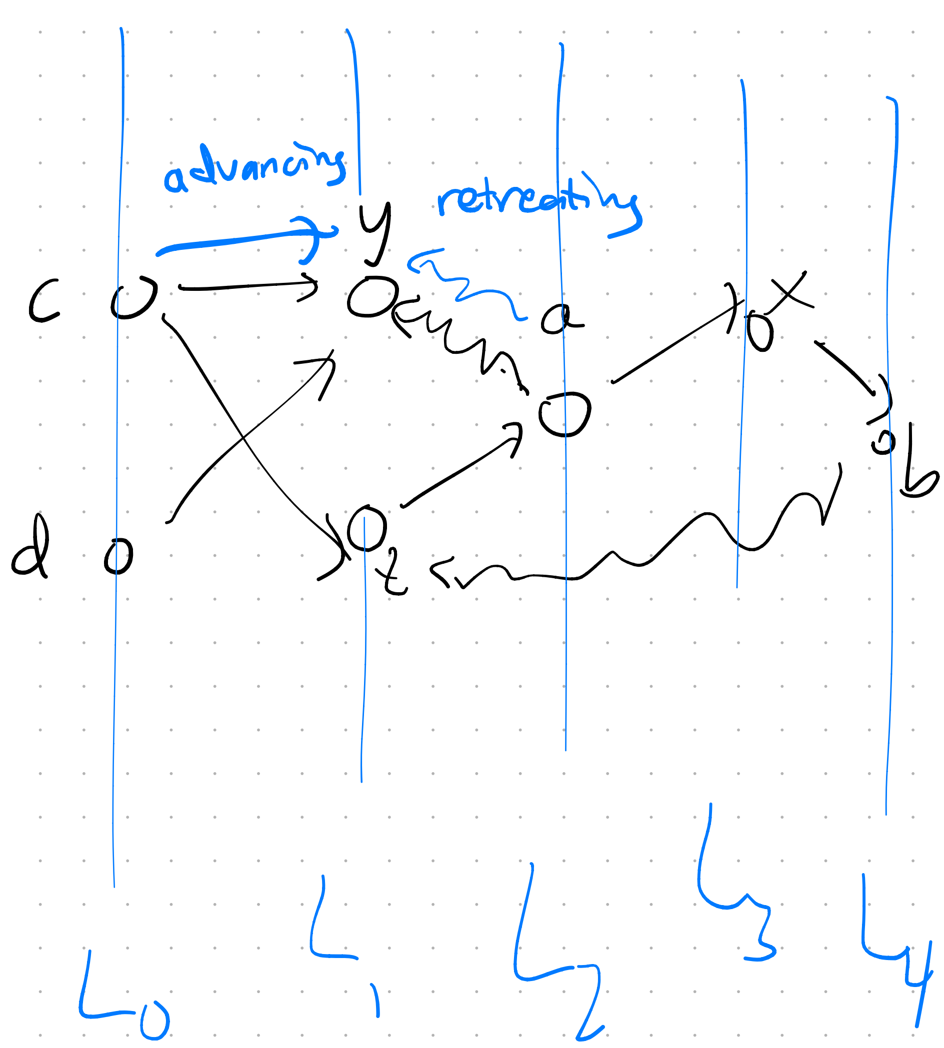
In the graph G_M, let $L_0, L_1, \ldots, L_\infty$ be the "BFS layers" where

$$L_0 = L \cap F$$

L_{i+1} = all vertices $v \notin L_0 \cup \ldots \cup L_i$ s.t. G_M contains an edge from L_i to v.

L_∞ = all vertices $v \notin \bigcup_{i=1}^{n} L_i$.





Def. A <u>blocking set</u> of M-augmenting paths is a (set-wise) maximal set of vertex-disjoint <u>advancing</u> M-augmenting paths. (Composed entirely of advancing edges.)

FACT. There is a linear-time algorithm to compute a blocking set of M-augmenting paths.
(Similar to DFS.)

H-K Algorithm

1. Initialize $M = \emptyset$.
2. While G contains an M-augmenting path:

   Let $P_1, \ldots, P_k$ be a blocking set.

   $M \leftarrow M \oplus (P_1 \cup \ldots \cup P_k)$

3. Output M.

Fact. (To be proven later)

\# while loop iterations $\le 2\sqrt{n}$,

$\Rightarrow \Theta(m\sqrt{n} + n)$ running time.