

Hopcroft Karp Algorithm for Bipartite Matching

CS 759 Perfect Matchings: Algorithms and Complexity

Riya Baviskar

Indian Institute of Technology, Bombay

April 12, 2019



Intuition behind the algorithm

The Augmenting path algorithm seen in class chooses one augmenting path in each iteration, even if it finds many augmenting paths in the process of searching.

The Hopcroft-Karp algorithm improves the running time of the above algorithm by correcting this wasteful aspect; in each iteration it attempts to find many disjoint augmenting paths, and it uses all of them to increase the size of matching.



Blocking set of augmenting paths

Definition

If G is a graph and M is a maximum matching, a blocking set of augmenting paths with respect to M is a set $\{P_1, \dots, P_k\}$ of augmenting paths such that:

- the paths P_1, \dots, P_k are vertex disjoint
- they all have the same length, l
- l is the minimum length of an M -augmenting path
- every augmenting path of length l has at least one vertex in common with $P_1 \cup \dots \cup P_k$



Lemma

Lemma

If M is a matching and P_1, \dots, P_k is any set of vertex-disjoint M -augmenting paths then $M \oplus P_1 \oplus P_2 \oplus \dots \oplus P_k$ is a matching of cardinality $|M| + k$.

($A \oplus B$ denotes the symmetric difference of two sets A and B , i.e. the set of all elements that belong to one of the sets but not the other)

Proof: Extension of $M \oplus P$.



Lemma

Lemma

Suppose G is a graph, M is a matching in G , and M^ is a maximum matching; let $k = |M^*| - |M|$. The edge set $M \oplus M^*$ contains at least k vertex-disjoint M -augmenting paths. Consequently, G has at least one M -augmenting path of length less than n/k , where n denotes the number of vertices of G .*



Proof

The edge set $M \oplus M^*$ has maximum degree 2, and each vertex of degree 2 in $M \oplus M^*$ belongs to exactly one edge of M . Therefore each connected component of $M \oplus M^*$ is an M -alternating component. Each M -alternating component which is not an augmenting path has at least as many edges in M as in M^* . Each M -augmenting path has exactly one fewer edge in M as in M^* . Therefore, at least k of the connected components of $M \oplus M^*$ must be M -augmenting paths, and they are all vertex-disjoint. Note that G has only n vertices, so it cannot have k disjoint subgraphs each with more than n/k vertices.



Algorithm

Hopcroft-Karp algorithm, outer loop

$M = \phi$

repeat

Let $\{P_1, \dots, P_k\}$ be a blocking set of augmenting paths
with respect to M

$M \leftarrow M \oplus P_1 \oplus P_2 \oplus \dots \oplus P_k$

until there is no augmenting path with respect to M



Lemma

Lemma

The minimum length of an M -augmenting path strictly increases after each iteration of the Hopcroft-Karp outer loop in which a non-empty blocking set of augmenting paths is found.



Proof

For graph $G(U,V,E)$,

Some notations:

M = matching at the start of one loop iteration

$\{P_1, \dots, P_k\}$ = blocking set of augmenting paths found

$Q = P_1 \cup \dots \cup P_k$

$R = E \setminus Q$

$M' = M \oplus Q$ = matching at the end of the iteration

F = vertices that are free with respect to M

F' = vertices that are free with respect to M'

$D(G,M)$ = directed graph formed from G by orienting each edge from U to V if it does not belong to M , and from V to U otherwise

$d(v)$ = length of shortest path in $D(G, M)$ from $U \cap F$ to v (If no such path exists, $d(v) = \infty$)



Proof continued

If (x, y) is any edge of $D(G, M)$ then $d(y) \leq d(x) + 1$. Edges of $D(G, M)$ that satisfy $d(y) = d(x) + 1$ will be called advancing edges, and all other edges will be called retreating edges. Note that Q is contained in the set of advancing edges.

In the edge set of $D(G, M')$, the orientation of every edge in Q is reversed and the orientation of every edge in R is preserved.

Therefore, $D(G, M')$ has three types of directed edges (x, y) :

- 1 reversed edges of Q , which satisfy $d(y) = d(x) - 1$
- 2 advancing edges of R , which satisfy $d(y) = d(x) + 1$
- 3 retreating edges of R , with satisfy $d(y) \leq d(x)$



Proof continued

Let l denote the minimum length of an augmenting path with respect to M , i.e. $l = \min\{d(v) \mid v \in V \cap F\}$.

Let P be any path in $D(G, M')$ from $U \cap F'$ to $V \cap F'$. The lemma asserts that P has at least l edges. The endpoints of P are free in M' , hence also in M . As w ranges over the vertices of P , the value $d(w)$ increases from 0 to at least l , and each edge of P increases the value of $d(w)$ by at most 1. Therefore P has at least l edges, and the only way that it can have l edges is if $d(y) = d(x) + 1$ for each edge (x, y) of P .



Proof continued

We have seen that this implies that P is contained in the set of advancing edges of R , and in particular P is edge-disjoint from Q . It cannot be vertex-disjoint from Q because then $\{P_1, \dots, P_k, P\}$ would be a set of $k + 1$ vertex-disjoint minimum-length M -augmenting paths, violating our assumption that $\{P_1, \dots, P_k\}$ is a blocking set. Therefore P has at least one vertex in common with $\{P_1, \dots, P_k\}$, i.e. $P \cap Q \neq \emptyset$. The endpoints of P cannot belong to Q , because they are free in M' whereas every vertex in Q is matched in M' . Let w be a vertex in the interior of P which belongs to Q . The edge of M' containing w belongs to P , but it also belongs to Q . This violates our earlier conclusion that P is edge-disjoint from Q , yielding the desired contradiction!



Lemma

Lemma

The Hopcroft-Karp algorithm terminates after fewer than $2\sqrt{n}$ iterations of its outer loop.

Proof.

After the first \sqrt{n} iterations of the outer loop are complete, the minimum length of an M -augmenting path is greater than \sqrt{n} . This implies $|M^*| - |M| < \sqrt{n}$, where M^* denotes maximum cardinality matching.

Each remaining iteration strictly increases $|M|$, hence there are fewer than \sqrt{n} iterations remaining. □



Final Complexity

The inner loop (finding blocking set of augmenting paths) can be done in $O(m)$ time.

Therefore, overall complexity of algorithm is $O(m\sqrt{n})$.



Thank You!

