Theorem 1.9

MBFS can be implemented such that it uses $O(m+n)$ time and $O(m+n)$ space.

Note that also with respect to $\bar{G}_M$, a simple path from s to t must not be strongly simple. Hence, we cannot compute a maximal set of up to s and t pairwise disjoint strongly simple paths from s to t using DFS.

Knowing $\bar{G}_M$ it is easy to compute a maximal set of shortest strongly simple paths using MDFS in $O(m+n)$ time. Everytime when a strongly simple path P from s to t is found, all nodes $[v,A]$, $[v,B]$ with $[v,A] \in P$ or $[v,B] \in P$ and all incident edges are deleted from $\bar{G}_M$.

If a node get zero indegree or zero outdegree then also this node and all incident edges are deleted. Note that after the deletion of a node and the incident edges, for strongly simple paths which are not destroyed by the deletion process, Lemma 1.2 remains valid.

Exercise

Work out the implementation of the Hopcroft-Karp approach for nonbipartite graphs.

1.3 The primal-dual method for weighted matching

Let $G = (V, E)$ be an undirected graph. If we associate with each edge $(i,j) \in E$ a weight $w_{ij} > 0$ then we obtain a weighted undirected graph $G = (V, E, w)$. The weight $w(M)$ of a matching M is the sum of the weights of the edges in M. A matching $M \subseteq E$ has maximum weight if $\sum_{(i,j) \in M'} w_{ij} \leq \sum_{(i,j) \in M} w_{ij}$ for all matchings $M' \subseteq E$. Given a weighted undirected graph $G = (V, E, w)$, the maximum weighted matching problem is finding a matching $M \subseteq E$ of maximum weight.

First, we will describe the primal-dual method for the computation of a maximum weighted matching. Let $G = (V, E, w)$ be a weighted undirected graph. Let

$$\mathcal{F} = \{E_1, E_2, \dots, E_r\}, \quad E_i \subseteq E$$

be a family of pairwise distinct subsets of E

With each node $i \in V$ we associate a

node weight $\pi(i) \geq 0$.

With each edge set $E_e \in \mathcal{F}$, we associate a

set weight $\mu(E_e) \geq 0$.

The new variables are called <u>dual variables</u>.

Note that the primal-dual method for bipartite graphs only uses dual variables with respect to the nodes of the graph. This suffices since every node $v \in A \cup B$ has the property that all M-alternating paths from an M-free node in B to v have even length if $v \in B$ and odd length if $v \in V$. But in nonbipartite graphs, with respect to a node $v \in V$ simultaneously, there can exist M-alternating paths from M-free nodes to v of odd and of even length. Moreover, both end nodes of an edge can have even or odd distances from the M-free nodes with respect to M-alternating paths. Hence, the dual variables with respect to the edge sets are needed. The exact reasons for this will be clearer during the development of the method.

The values of the dual variables are treated such that the following invariant is always fulfilled.

- For all $(i,j) \in E$ there hold

$$w_{ij} \leq \pi(i) + \pi(j) + \sum_{(i,j) \in E_e} \mu(E_e).$$

For each edge $(i,j) \in E$, its <u>dual weight</u> $d(i,j)$ is defined by

$$d(i,j) = \pi(i) + \pi(j) + \sum_{(i,j) \in E_e} \mu(E_e).$$

We define the <u>dual weight</u> $d(M)$ of a matching M by

$$d(M) = \sum_{(i,j) \in M} d(i,j).$$

Note that always for all matchings $M \subseteq E$

$$w(M) \leq d(M).$$

With respect to an arbitrary matching $M \subseteq E$, the maximum contribution of the node weight $\pi(i)$ to its dual weight can be $\pi(i)$ since i is adjacent to at most one edge in M.

Let $c(E_e)$ be the size of a maximum cardinality matching with respect to E_e.

Note that $|E_e \cap M| \leq c(E_e)$. Hence, the maximum contribution of the set weight $\mu(E_e)$ can be $c(E_e) \cdot \mu(E_e)$. Hence,

$$\sum_{i \in V} \pi(i) + \sum_{E_e \in F} c(E_e) \mu(E_e)$$

will be always an upper bound for the dual weight of any matching of G.

Therefore, with respect to a matching M

$$w(M) = \sum_{i \in V} \pi(i) + \sum_{E_e \in \mathcal{F}} c(E_e)\,\mu(E_e)$$

implies that the matching M is a maximum weighted matching.

The question is now:

When with respect to a matching M, this equality holds?

Since the dual weight of an edge is at least as large as its weight, we obtain the necessary condition

$$d(i,j) = w(i,j)$$

We ~~wr~~ write also $w(i,j)$ for the weight w_{ij}.

for all edges $(i,j) \in M$.

Since all summonds in both sums are non-negative, the node weight $\pi(i)$ has to be 0 for all M-free nodes $i \in V$.

For all E_e such that $|M \cap E_e| < c(E_e)$ the set weight $\mu(E_e)$ has to be 0.

Altogether, we obtain the following necessary and sufficient conditions:

1. $r(i,j) := d(i,j) - w(i,j) = 0 \quad \forall (i,j) \in M$,

2. $\pi(i) = 0 \quad \forall$ M-free $i \in V$, and

3. $\mu(E_e) = 0 \quad \forall E_e \in \mathcal{F}$ with $|E_e \cap M| < c(E_e)$.

The value $r(i,j)$ is called the <u>reduced cost</u> of the edge (i,j).

The primal-dual method for the weighted matching problem can be separated into rounds. The input of every round will be a matching M and values for the dual variables which fulfill the Conditions 1 and 3 with respect to the matching M.

Our goal within a round is to modify M and the values of the dual variables such that Conditions 1 and 3 remain valid and the number of nodes violating Condition 2 is strictly decreased

One round devides into two steps, the <u>search step</u> and the <u>extension step</u>.

The search step try to improve the current matching by finding an augmenting path P such that the number of free nodes with node weight larger than 0 can be decreased by the augmentation of P. Since Condition 1 has to be maintained, the search step can only be performed on edges with reduced cost 0.

If this is not possible then the extension step changes the values of some dual variables by an appropriate value δ. The extension step can decrease the reduced cost of some edges to 0. Hence, the next search step possibly finds an augmenting path.

During the search step, we use MDFS. Hence, we define with respect to the current matching M the weighted directed bipartite graph $G_M = (V', E_M, w)$ as follows:

$$V' := \{[v,A], [v,B] \mid v \in V\} \cup \{s,t\} \quad s,t \notin V, \; s \neq t$$

$$\begin{aligned} E_M := & \{([v,A],[w,B]), ([w,A],[v,B]) \mid (v,w) \in M\} \\ & \cup \{([x,B],[y,A]), ([y,B],[x,A]) \mid (x,y) \in E \setminus M\} \\ & \cup \{(s,[v,B]), ([v,A],t) \mid v \in V \text{ is } M\text{-free}\}. \end{aligned}$$

Both copies $([i,X],[j,\bar{X}])$ and $([j,X],[i,\bar{X}])$, $X \in \{A,B\}$ of the edge (i,j) obtain

weight $w(i,j)$ and reduced cost $r(i,j)$.

We arrange that edges with tail s or head t have always reduced cost 0.

According to Condition 1, it is only allowed to consider augmenting paths where all edges on these paths have reduced cost 0.

$\Rightarrow$

The input graph $G_M^* = (V', E_M^*, w)$ will be the subgraph of G_M containing exactly those edges in E_M having reduced cost 0. I.e.,

$$E_M^* := \{([i,x],[j,\bar{x}]) \in E_M \mid r(i,j) = 0\}$$

Condition 1 $\Rightarrow$ $M \subseteq E_M^*$.

We start with the empty matching $\emptyset$ and define the graph $G_\emptyset = (V', E_\emptyset, w)$ as described above. Let

$$W := \max\{w(i,j) \mid (i,j) \in E\}.$$

We initialize all node weights $\pi(i)$ by

$$\pi(i) := \frac{W}{2} \quad \text{for all } i \in V.$$

At the beginning, we have

$$\mathcal{F} = \emptyset$$

such that no set weight has to be defined.

In dependence to the algorithm, the needed elements of $\mathcal{F}$ and the corresponding set weights will be defined.

As soon as $\mu(E_e)$ becomes zero for an edge set $E_e \in \mathcal{F}$, we will delete E_e from $\mathcal{F}$.

$\leadsto$

Input graph $G_\phi^* = (V', E_\phi^*, w)$ for the first search ~~step~~ where

$$E_\phi^* = \{([i,B],[i,A]),([j,B],[i,A]) \mid ([i,B],[j,A]) \in E_\phi \text{ and } w(i,j) = W\}$$

$$\cup \{(s,[i,B]),([i,A],t) \mid i \in V\}.$$

A search step terminates with

- a matching M,
- a weighted directed graph $G_M = (V', E_M, w)$, and
- a current subgraph $G_M^* = (V', E_M^*, w)$ of G_M such that no M-augmenting path P is contained in G_M^*.

This is the input for the extension step.

~~For~~ the treatment of the extension step consider the expanded MDFS-tree T_{exp} computed by the last MDFS on G_M^*.

Note that this MDFS was unsucessful; i.e., no path from the start node s to the target node t was found.

Goal:

To add edges to T_{exp} such that possibly an augmenting path

$\Longrightarrow$

We have to decrease the reduced cost of edges with positive reduced cost.

Properties:

- Such edge (i,j) has to be in $E \setminus M$ and
- $[i,B]$ has to be a node in T_{exp}.

Let $V_A = \{ [i,A] \mid [i,A] \in V' \}$ and

$V_B = \{ [i,B] \mid [i,B] \in V' \}$.

$A_T := V_A \cap T_{exp}$ and for a moment

$B_T := V_B \cap T_{exp}$.

Later, we shall see that according to the conditions which we have to maintain, some nodes in $V_B \cap T_{exp}$ are not allowed to be a node in B_T.

Idea:

Decrease the reduced cost $r(i,j)$ of all edges (i,j) with positive reduced cost and $[i,B] \in B_T$ by the appropriate value δ.

With respect to the other end node j of edge (i,j), the following four cases can arise:

1. $[j,B] \notin B_T$ and $[j,A] \notin A_T$,
2. $[j,B] \notin B_T$ and $[j,A] \in A_T$,
3. $[j,B] \in B_T$ and $[j,A] \notin A_T$, and
4. $[j,B] \in B_T$ and $[j,A] \in A_T$.

We decrease $r(i,j)$ by decreasing $\pi(i)$ by the appropriate value δ; i.e., we

- decrease $\pi(i)$ by δ for all nodes i with $[i,B] \in B_T$.

$\Rightarrow$ The reduced cost of edges e in G_M^* with end node $[i,A]$ or $[i,B]$ becomes negative.

Condition 1 $\Rightarrow$

We have to increase such reduced cost.

We distinguish two cases:

1. The other end node of e corresponds to A_T but not to B_T.

2. The other end node corresponds to B_T.

Case 1:

Then we can increase the reduced cost of edge e by increasing the node weight of the other end node of e by δ; i.e., we increase $\pi(j)$ by δ for all nodes j such that $[j, A] \in A_T$ and $[j, B] \notin B_T$.

Case 2:

Then the reduced cost $r(i,j)$ is decreased by 2δ. This can be corrected by increasing the set weight $\mu(E_e)$ of exactly one set E_e containing the edge (i,j) by 2δ.

E_e will have the property that all edges in E_e are in E_M^* and both end nodes are contained in B_T.

Questions:

1. What is the accurate edge set E_e for increasing its set weight?

2. What is the appropriate value δ?

To answer the first question let us consider MDFS which is used as subroutine during the search step. Review the definitions and the properties of the sets $L_{[w,A]}$ and $D_{[q,A]}$ as given during the development of MDFS.

First, it is useful to investigate the structure of a set $D_{[q,A]}$. Let

$$D'_{[q,A]} := \{ p \in V \mid [p,A] \in D_{[q,A]} \} \cup \{[q,A]\}.$$

Furthermore, let

$$\tilde{D}_{[q,A]} := \{ [p,A], [p,B] \mid p \in D'_{[q,A]} \}.$$

The unique node $p \in D'_{[q,A]}$ such that p is end node of an edge $(r,p) \in M$ with $r \notin D'_{[q,A]}$ is the node q. Let $(r,q] \in M$ be the unique matched edge with end node q.

We say that a path P <u>enters</u> or <u>leaves</u> $\tilde{D}_{[q,A]}$ <u>via an edge $([x,B],[y,A])$ in $E \setminus M$</u> if $(x,y) \in E \setminus M$

During the performance of MDFS, for an M-augmenting path P there are three possibilities to run through a set $\tilde{D}_{[q,A]}$:

1. P enters and leaves $\tilde{D}_{[q,A]}$ via an edge in $E \setminus M$.

2. P enters $\tilde{D}_{[q,A]}$ via the matched edge $([r,A],[q,B])$ and leaves $\tilde{D}_{[q,A]}$ via an edge in $E \setminus M$.

3. P enters $\tilde{D}_{[q,A]}$ via an edge ~~in~~ $E \setminus M$ and leaves $\tilde{D}_{[q,A]}$ via the matched edge $([q,A],[r,B])$.

If an M-alternating path R enters $\tilde{D}_{[q,A]}$ via the edge $([r,A],[q,B])$ then, by Lemma 1.3, for all $v \in D'_{[q,A]}$, $[v,B] \in \mathcal{B}_T$.

Then with respect to each edge in

$$\hat{E}^*_M := \{(i,j) \mid ([i,A],[j,B]) \in E^*_M \text{ or } ([i,B],[j,A]) \in E^*_M\}$$

with both end nodes in $D'_{[q,A]}$, we have to increase exactly one edge set containing this edge.

Note that for all $v \in V$ there exists at most one current $D_{[q,A]}$ such that $v \in D'_{[q,A]}$. Hence, we define the edge set E_q corresponding to $D'_{[q,A]}$ by

$$E_q := (D'_{[q,A]} \times D'_{[q,A]}) \cap \hat{E}^*_M .$$

Note that E_q changes when $\hat{E}^*_M$ changes.