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 $v \in D_{[q,A]}$, $[v,B] \in B_T$.

Then with respect to each edge in E with both end nodes in $D'_{[q,A]}$, we have to increase the set weight of 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

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

If we have to increase the set weight with respect to an edge $(i,j) \in E$ then we choose the edge set E_q where $D_{[q,A]}$ is the current set with the property that $i,j \in D'_{[q,A]}$. Note that $[i,B], [j,B] \in B_T$ and $(i,j) \in E$ implies that $D_{[q,A]}$ exists.

<u>Exercise:</u>

Prove that $[i,B], [j,B] \in \mathcal{B}_1$ and $(i,j) \in E$ implies that there exists $D'_{[q,A]}$ current such that $i, j \in D'_{[q,A]}$.

Let us examine the effect of the augmentation of P to the number of edges in the current matching with both end nodes in $D'_{[q,A]}$.

- If the crossing of P through $\tilde{D}_{[q,A]}$ is of Type 1 then this number decreases by 1. In the other two cases, this number does not change.

$\Longrightarrow$

The augmentation of an augmenting path of Type 2 or 3 is always allowed but the augmentation of an augmenting path of Type 1 is only allowed if $\mu(E_q) = 0$; i.e., E_q has left $\mathcal{F}$.

Next we shall determine the accurate value for δ.

Since all node weights have to be nonnegative, δ cannot be larger than the node weight of an M-free node i. Note that with respect to an M-free node i, always $[i,B] \in \mathcal{B}_1$ is fulfilled.

$\Longrightarrow$

All free nodes have the same node weight.

Hence, $\delta \leq \pi(i)$, i M-free implies that after the change of the dual values all node weights are nonnegative.

Remember that for $[i,B] \in B_T$, we have separated the edges $([i,B],[j,A])$ into four classes.

- If edge $([i,B],[j,A])$ is of Type 1, we have to choose $\delta = r(i,j)$ for decreasing the reduced cost $r(i,j)$ of the edge (i,j) to 0.
- If edge $([i,B],[j,A])$ is of Type 2, independently from the choice of δ, $r(i,j)$ does not change.
- If edge $([i,B],[j,A])$ is of Type 3 or 4 then we have to choose $\delta = \frac{1}{2} r(i,j)$ for decreasing $r(i,j)$ to 0 since both node weights $\pi(i)$ and $\pi(j)$ will be decreased.

Note that δ has to be chosen in such a way that after the extension step $r(i,j) \geq 0$ for all edges $(i,j) \in E$.

$\Rightarrow$

δ should not be larger than the minimal reduced cost with respect to edges (i,j) with

$[i,B] \in B_T$ and $[j,A] \notin A_T$ and also not larger than the half of the minimal reduced cost with respect to edges (i,j) with $[i,B], [j,B] \in B_T$, $\nexists$ $D_{[q,A]}$ current with $i,j \in D_{[q,A]}$ and $r(i,j) > 0$.

Exercise:

Discuss the case that we have an edge $(i,j) \in E$ with $[i,B], [j,B] \in B_T$ and $r(i,j) = 0$.

Since Condition 3 has to be maintained with respect to current $D_{[q,A]}$ the following holds:

If during the MDFS no path enters $\tilde{D}'_{[q,A]}$ via $([r,A],[q,B])$ but there is a path R entering $\tilde{D}'_{[q,A]}$ via an edge in $E \setminus M$ then $\mu(E_q) > 0$ implies that R has to leave $\tilde{D}'_{[q,A]}$ via $([q,A],[r,B])$, independently if $[q,A]$ is already pushed or not. Since $(q,r) \in M$ and therefore $([q,A],[r,B]) \in E_M^+$, this is always possible. Note that R enters $\tilde{D}'_{[q,A]}$ through a node in A_T and leaves $\tilde{D}'_{[q,A]}$ through a node in A_T.

In dependence which nodes in $D_{[q,A]}$ are entering nodes of such paths R, with respect to a node $v \in D'_{[q,A]}$ the following can happen:

a) $[v,B] \in B_T$,
b) $[v,B] \notin B_T$ but $[v,A] \in A_T$, and

c) $[v,B] \notin B_T$ and $[v,A] \notin A_T$.

Problem: How to change the node weights of the nodes in $D'_{[q,A]}$?

Each entering node $[v,A]$ of $D_{[q,A]}$ is the head of an edge $([x,B],[v,A])$ where $[x,B] \in B_T$. Hence, $\pi(x)$ is decreased by δ. According to Condition 1, $\pi(v)$ has to be increased by δ.

Possibly there are edges in E_q with exactly one end node is an entering node, with both end nodes are entering nodes or with no end node is an entering node. With respect to all these cases, the node weights and $\mu(E_q)$ have to be changed in such a way that the three conditions re= main to be valid. For doing this, we

- increase $\pi(v)$ by δ for all $v \in D'_{[q,A]}$.

Since we have increased the reduced cost of all edges in E_q by 2δ, we

- decrease $\mu(E_q)$ by 2δ.

Since $\mu(E_q)$ has to be nonnegative, δ has to be chosen such that before the change of the dual variables $\mu(E_q) \geq 2\delta$.

25.06.

Remember that we consider the situation that $[q,B] \notin T_{exp}$. Possibly, there exist nodes

$[i,B] \in \tilde{D}_{[q,A]}$ which are also a node in T_{exp}. Since we cannot decrease $\pi(i)$ by δ and increase $\pi(i)$ by δ at the same time, it is not allowed for such a node to be in B_T. Hence, we define

$$B_f := \{ [i,B] \mid [i,B] \in \tilde{D}_{[q,A]},\ D_{[q,A]} \text{ current and } [q,B] \in T_{exp} \}.$$

and redefine

$$B_T := (V_B \cap T_{exp}) \setminus B_f .$$

Altogether, we can define δ in the following way:

$$\delta_0 := \pi(i) \qquad \text{where } i \text{ is } M\text{-free},$$

$$\delta_1 := \min \{ r(i,j) \mid [i,B] \in B_T \text{ and } [j,A] \notin A_T \},$$

$$\delta_2 := \min \{ r(i,j) \mid [i,B],[j,B] \in B_T,\ r(i,j) > 0 \text{ and } \nexists D_{[q,A]} \text{ current with } i,j \in D'_{[q,A]} \},$$

and

$$\delta_3 := \min \{ \mu(E_q) \mid D_{[q,A]} \text{ current}, [q,B] \notin B_T \text{ and } [q,A] \in A_T \}$$

Then we define

$$\delta := \min \{ \delta_0, \delta_1, \frac{\delta_2}{2}, \frac{\delta_3}{2} \}.$$

Altogether, we have obtained the following extension step:

$\delta_0 := \pi(i)$ for an M-free node i;

$\delta_1 := \min\{ r(i,j) \mid [i,B] \in \mathcal{B}_T$ and $[j,A] \notin A_T \}$;

$\delta_2 := \min\{ r(i,j) \mid [i,B], [j,B] \in \mathcal{B}_T,\ r(i,j) > 0$ and $\nexists\, D_{[q,A]}$ current with $i,j \in D'_{[q,A]} \}$;

$\delta_3 := \min\{ \mu(E_q) \mid D_{[q,A]}$ current, $[q,B] \notin \mathcal{B}_T$ and $[q,A] \in A_T \}$

$\delta := \min\{ \delta_0, \delta_1, \frac{\delta_2}{2}, \frac{\delta_3}{2} \}$;

<u>for</u> all $[i,B] \in \mathcal{B}_T$
  <u>do</u>
    $\pi(i) := \pi(i) - \delta$
  <u>od</u>;

<u>for</u> all $([i,B] \notin \mathcal{B}_T, [i,A] \in A_T)$ <u>or</u> $i \in D_{[q,A]}$, $D_{[q,A]}$ current with $[q,B] \notin \mathcal{B}_T$ but $[q,A] \in A_T$
  <u>do</u>
    $\pi(i) := \pi(i) + \delta$
  <u>od</u>;

<u>for</u> all $D_{[q,A]}$ current and $[q,B] \in \mathcal{B}_T$
  <u>do</u>
    $\mu(E_q) := \mu(E_q) + 2\delta$
  <u>od</u>;

for all $D_{[q,A]}$ current, $[q,B] \notin B_T$ and $[q,A] \in A_T$
do
$$\mu(E_q) := \mu(E_q) - 2\delta$$
od.

Exercise:
Prove the correctness of the described primal-dual method for the maximum weighted matching problem.

1.4 An implementation of the primal-dual method

First, we will determine an upper bound for the number of dual changes which can occur between two augmentations in the worst case. We distinguish four cases.

Case 1: $\delta = \delta_0$

Then after the change of the dual variables, $\pi(i) = 0$ for all M-free nodes $i \in V$. Hence, the current matching M is of maximum weight and the algorithm terminates. Hence, Case 1 occurs at most once.

Case 2: $\delta = \delta_1$

Then during the next search step at least one new node $[j, A]$ enters A_T. Hence, Case 2 occurs at most n times.

Case 3: $\delta = \delta_2$

Then, during the next search step, at least one new edge enters E_M^*. Furthermore, there is an union of the two distinct current sets $D_{[q_1, A]}$ and $D_{[q_2, A]}$ with one end node of the new edge is in $D'_{[q_1, A]}$ and the other end node is in $D'_{[q_2, A]}$ obtaining a new larger current set.

$\Rightarrow$

Case 3 occurs at most n times.

Case 4: $\delta = \delta_3$

Then at least one set E_q leaves the family $\mathcal{F}$. As long as $[q, B] \notin B_T$, no edge set E_q can enter the family $\mathcal{F}$. But if $[q, B]$ is pushed, a corresponding edge set E_q cannot contribute to the definition of δ_3 before $[q, B]$ leaves B_T again. This cannot happen before the next augmentation.

$\Rightarrow$

Case 4 occurs at most n times.

First, we will discuss the implementation of the search step between two augmentations.

Note that after an extension step, the last MDFS can be continued instead of to start a new MDFS. With respect to the primal-dual method, the following special situation has to be treated by the search step:

If according to an extension step, $\mu(E_q)$ becomes zero, the corresponding edge set E_q leaves the family $\mathcal{F}$. Now it is allowed that an augmenting path P enters and leaves $D_{[q,A]}$ via an edge in $E \setminus M$ but possibly not everywhere.

Maybe there is a set $D_{[q',A]} \subset D_{[q,A]}$ with $\mu(E_{q'}) > 0$. Hence, we have to undo the union operation done with respect to $D_{[q,A]}$. Note that this union operation has been performed before the last augmentation. Since $[q,A]$ is already pushed, no set $D_{[q,A]}$ can become current before the next augmentation.

Goal:

Extension of the data structure which uses the weighted union heuristic such that the time used for the deunion operations will be, up to a small constant factor, the same as the time used for the union operations and each find operation uses only constant time.

This can be done in the following way:

During the performance of an union operation, instead of changing a pointer, we add a new pointer. The current pointer of an element will be always the last created pointer.

Construction ⟹

We use for each element at most $\log n$ extra pointers.

The time used for the union operations remains essentially the same.

A deunion can be performed by the deletion of the current pointers created during the corresponding union operation and the update of the set sizes and of the name of the larger subset.

Exercise

Give an implementation of the data structure for union-find-deunion such that the time used for the operations is $O(m + n \cdot \log n)$.

Next, we shall give an implementation of the computation of the δ's and the update of the dual variables.