Chapter 6
Graph Algorithms

Algorithm Theory
WS 2016/17

Fabian Kuhn

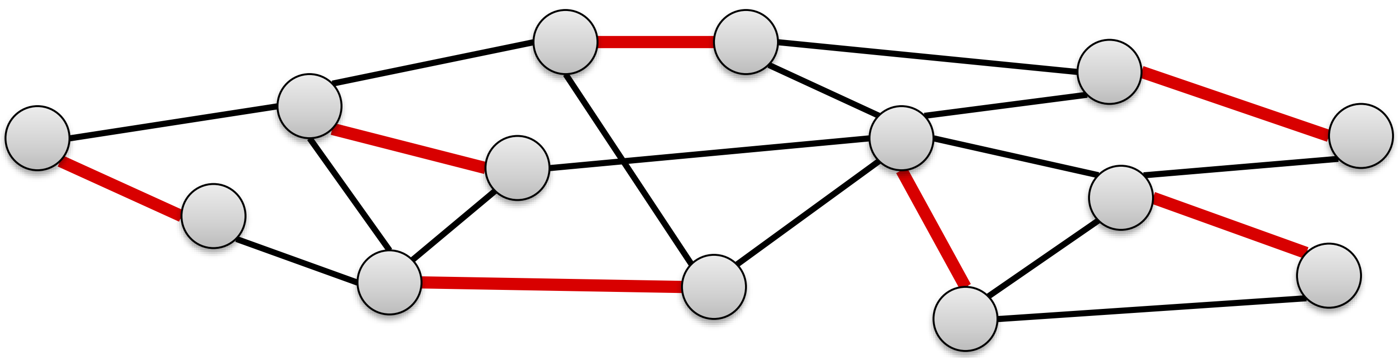
Vertex Cover vs Matching

Consider a matching M and a vertex cover S

Claim: $|M| \leq |S|$ for bipartite graphs : $|M^*| = |S^*|$

Proof:

- At least one node of every edge $\{u, v\} \in M$ is in S
- Needs to be a different node for different edges from M

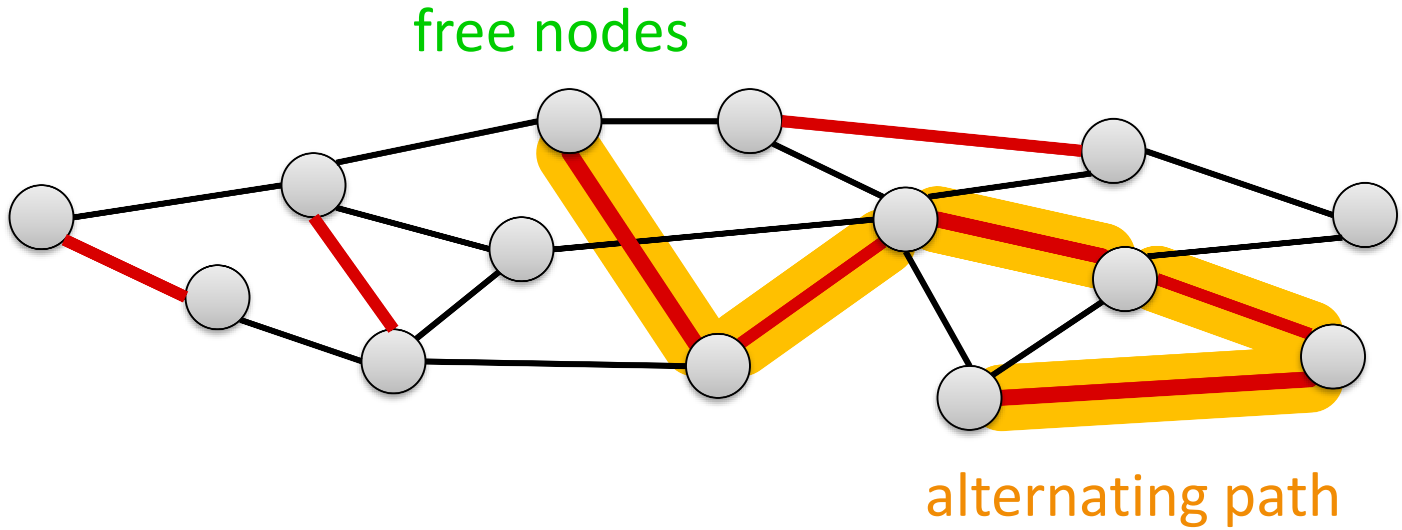


Augmenting Paths

Consider a matching M of a graph $G = (V, E)$:

- A node $v \in V$ is called **free** iff it is not matched

Augmenting Path: A (odd-length) path that starts and ends at a free node and visits edges in $E \setminus M$ and edges in M alternatingly.



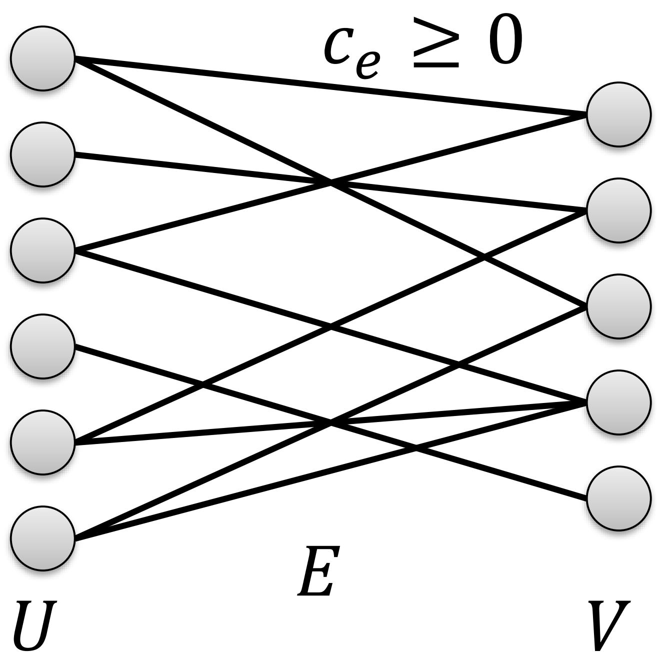
- Matching M can be improved using an augmenting path by switching the role of each edge along the path

Maximum Weight Bipartite Matching

- Let's again go back to bipartite graphs…

Given: Bipartite graph $G = (U \dot{\cup} V, E)$ with edge weights $c_e \geq 0$

Goal: Find a matching M of maximum total weight

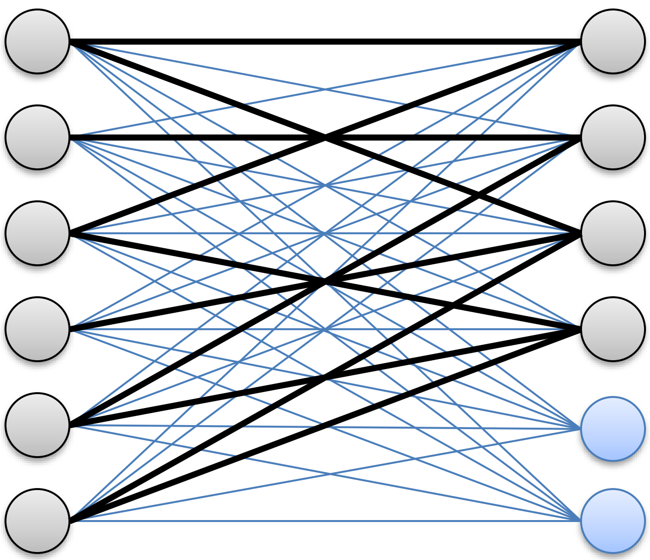


Minimum Weight Perfect Matching

Claim: Max weight bipartite matching is equivalent to finding a minimum weight perfect matching in a complete bipartite graph.

1. Turn into maximum weight perfect matching
   - add dummy nodes to get two equal-sized sides
   - add edges of weight 0 to make graph complete bipartite
2. Replace weights: $c_e' := \max_f\{c_f\} - c_e$

$\frac{n}{2}$



$\frac{n}{2}$

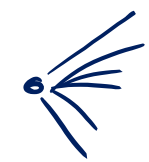
As an Integer Linear Program

- We can formulate the problem as an integer linear program

Var. x_{uv} for every edge $(u, v) \in U \times V$ to encode matching M:

$$x_{uv} = \begin{cases} 1, & \text{if } \{u, v\} \in M \\ 0, & \text{if } \{u, v\} \notin M \end{cases}$$

Minimum Weight Perfect Matching

u 

$$\min \sum_{(u,v) \in U \times V} c_{uv} \cdot x_{uv}$$

s.t.

$$\forall u \in U: \sum_{v \in V} x_{uv} = 1, \qquad \forall v \in V: \sum_{u \in U} x_{uv} = 1$$

$$\forall u \in U, \forall v \in V: \; x_{uv} \in \{0,1\}$$

$x_{uv} \geq 0$

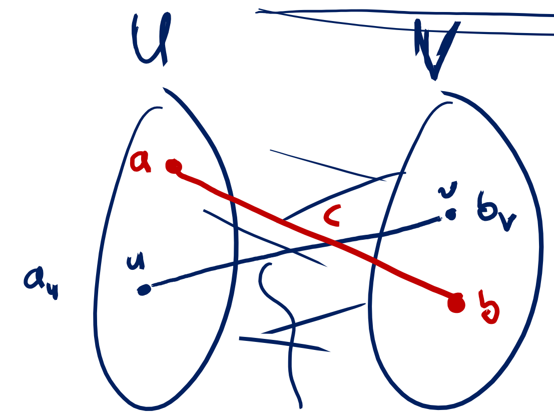
Dual Problem

- Every linear program has a dual linear program
  - The dual of a minimization problem is a maximization problem
  - Strong duality: primal LP and dual LP have the same objective value

In the case of the minimum weight perfect matching problem

- Assign a variable $a_u \geq 0$ to each node $u \in U$
  and a variable $b_v \geq 0$ to each node $v \in V$
- **Condition:** for every edge $(\boldsymbol{u}, \boldsymbol{v}) \in \boldsymbol{U} \times \boldsymbol{V}$: $\boldsymbol{a_u + b_v \leq c_{uv}}$
- Given perfect matching M:

$$\boxed{\sum_{(u,v)\in M} c_{uv} \geq \sum_{u\in U} a_u + \sum_{v\in V} b_v}$$



$c_{uv} \geq a_u + b_v$

$c_1 \geq a_1 + b_1$
$c_2 \geq a_2 + b_2$
$\vdots$
$c_{n/2} \geq a_{n/2} + b_{n/2}$

Dual Linear Program

- Variables $a_u \geq 0$ for $u \in U$ and $b_v \geq 0$ for $v \in V$

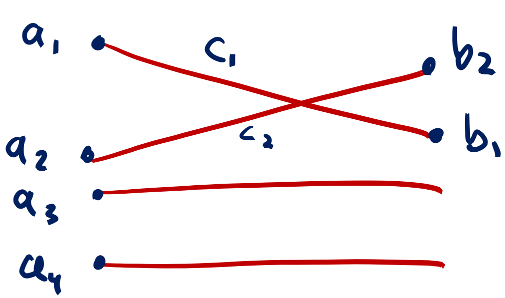
$$\max \sum_{u \in U} a_u + \sum_{v \in V} b_v$$

$$s.t.$$

$$\forall u \in U, \forall v \in V: \boxed{a_u + b_v \leq c_{uv}}$$

- For every perfect matching M: (for every feasible dual sol.)

$$\sum_{(u,v) \in M} c_{uv} \underset{=}{\geq} \sum_{u \in U} a_u + \sum_{v \in V} b_v$$



for all $(u,v) \in M$:

$c_{uv} = a_u + b_v$

$w_{uv} = c_{uv} - a_u - b_v = 0$

Complementary Slackness

- A perfect matching M is optimal if

$$\sum_{(u,v)\in M} c_{uv} = \sum_{u\in U} a_u + \sum_{v\in V} b_v$$

$C_{uv} \geq a_u + b_v$

- In that case, for every $(u,v) \in M$

$$w_{uv} := c_{uv} - a_u - b_v = 0$$

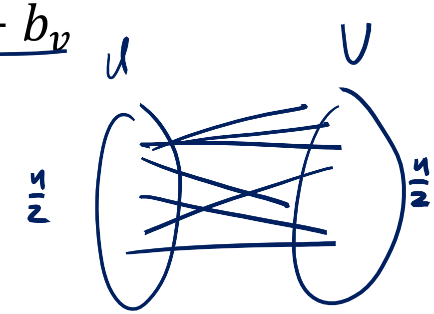
  – In this case, M is also an optimal solution to the LP relaxation of the problem
  – Every optimal LP solution can be characterized by such a property, which is then generally referred to as complementary slackness

- **Goal:** Find a dual solution a_u, b_v and a perfect matching such that the complementary slackness condition is satisfied!
  – i.e., for every matching edge (u,v), we want $w_{uv} = 0$
  – We then know that the matching is optimal!

Algorithm Overview

- Start with any feasible dual solution a_u, b_v
  - i.e., solution satisfies that for all (u, v): $c_{uv} \geq a_u + b_v$

    for example: $a_u = 0$, $b_v = 0$ $\forall u, v$

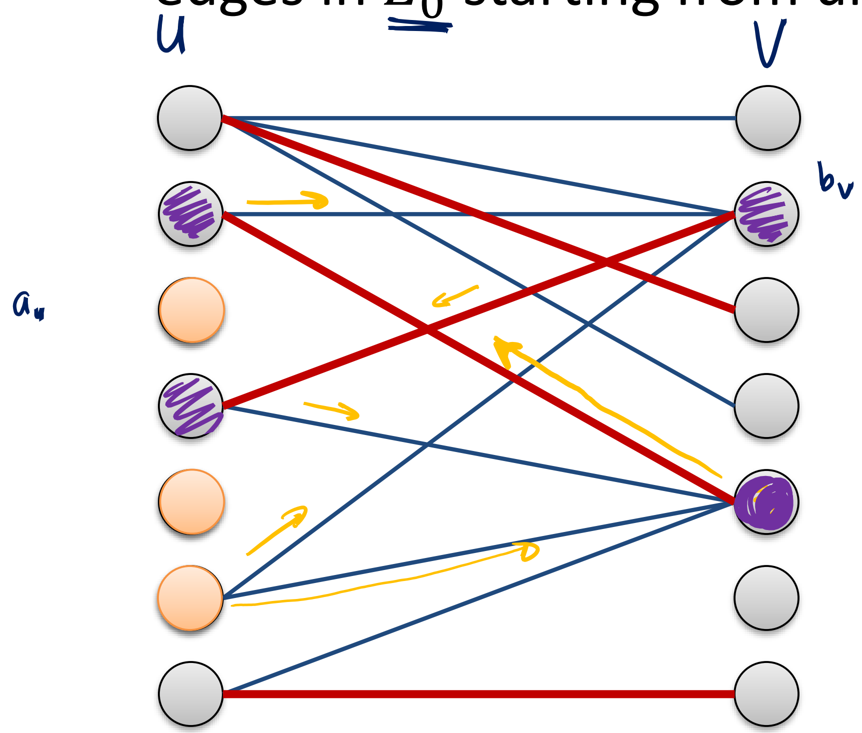


- Let E_0 be the edges for which $w_{uv} = 0$
  - Recall that $w_{uv} = c_{uv} - a_u - b_v$
- Compute maximum cardinality matching M of E_0
- All edges (u, v) of M satisfy $w_{uv} = 0$
  - Complementary slackness if satisfied
  - If M is a perfect matching, we are done ✓
- If M is not a perfect matching, dual solution can be improved

Marked Nodes

Define set of marked nodes L:

- Set of nodes which can be reached on alternating paths on edges in E_0 starting from unmatched nodes in U



edges E_0 with $w_{uv} = 0$

optimal matching M

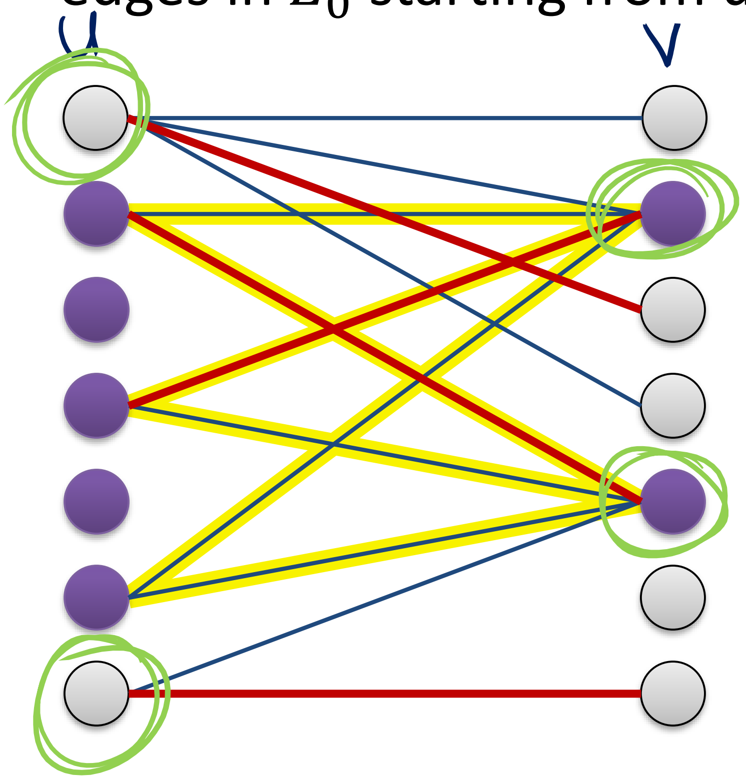
L_0: unmatched nodes in U

L: all nodes that can be reached on alternating paths starting from L_0

Marked Nodes

Define set of marked nodes L:

- Set of nodes which can be reached on alternating paths on edges in E_0 starting from unmatched nodes in U



edges E_0 with $w_{uv} = 0$

optimal matching M

L_0: unmatched nodes in U

L: all nodes that can be reached on alternating paths starting from L_0

Observation:

all marked nodes in V are matched.

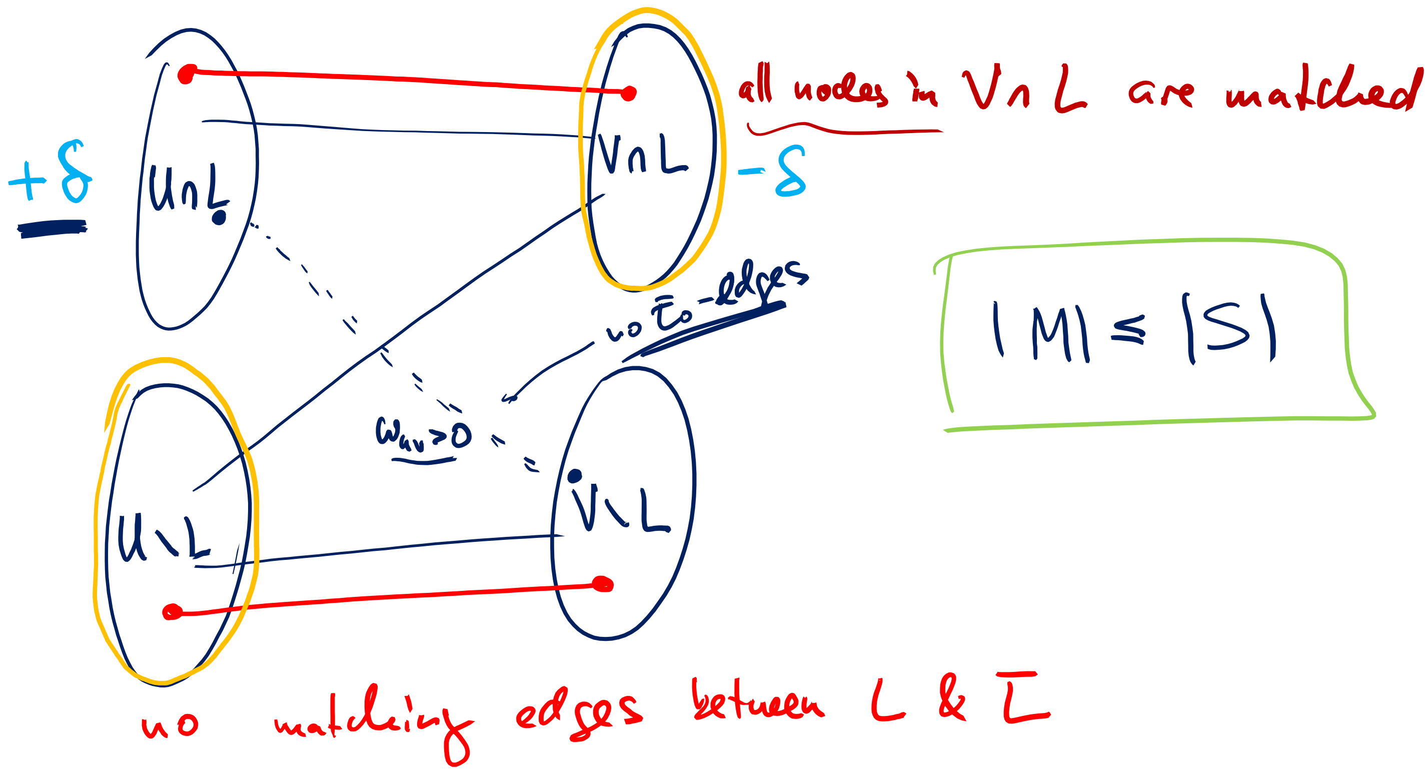
→ otherwise: augm. path

Marked Nodes – Vertex Cover

Lemma:

a) There are no E_0-edges between $U \cap L$ and $V \setminus L$

b) The set $(U \setminus L) \cup (V \cap L)$ is a vertex cover of size $|M|$ of the graph induced by E_0



$|M| \le |S|$

Improved Dual Solution

Recall: all edges (u, v) between $U \cap L$ and $V \setminus L$ have $w_{uv} > 0$

New dual solution:

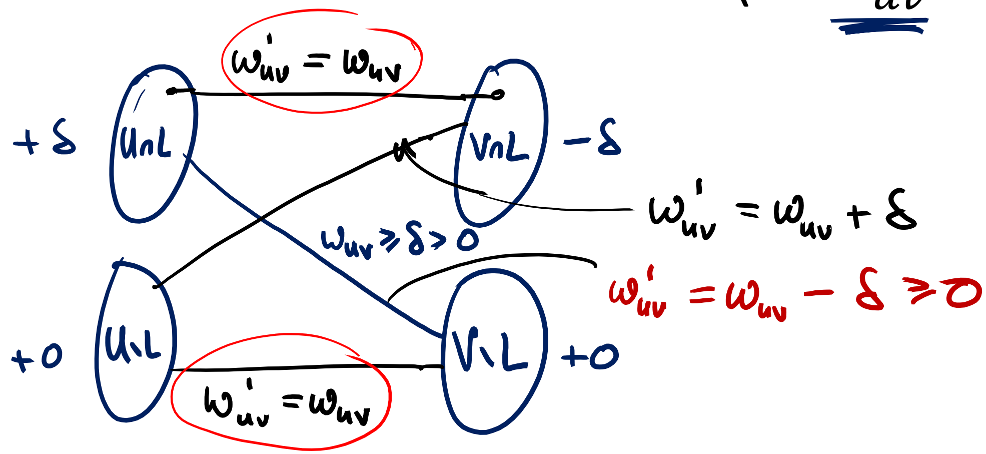
$w_{uv} = c_{uv} - a_u - b_v$

$$\delta := \min_{u \in U \cap L,\, v \in V \setminus L} \{w_{uv}\}$$

$$a'_u := \begin{cases} a_u, & \text{if } u \in U \setminus L \\ a_u + \delta, & \text{if } u \in U \cap L \end{cases}$$

$$b'_v := \begin{cases} b_v, & \text{if } v \in V \setminus L \\ a_v - \delta, & \text{if } v \in V \cap L \end{cases}$$

Claim: New dual solution is feasible (all w_{uv} remain ≥ 0)



Improved Dual Solution

Lemma: Obj. value of the dual solution grows by $\delta\left(\underbrace{\frac{n}{2} - |M|}_{>0}\right)$.

Proof: matching M not perfect $\Longleftrightarrow$ $|M| < \frac{n}{2}$

$$\delta := \min_{u \in U \cap L,\, v \in V \setminus L} \{w_{uv}\}, \qquad a'_u := \begin{cases} a_u, & \text{if } u \in U \setminus L \\ a_u + \delta, & \text{if } u \in U \cap L \end{cases}, \qquad b'_v := \begin{cases} b_v, & \text{if } v \in V \setminus L \\ a_v - \delta, & \text{if } v \in V \cap L \end{cases}$$

$$D = \sum_{u \in U} a_u + \sum_{v \in V} b_v \qquad D' = \sum_{u \in U} a'_u + \sum_{v \in V} b'_v$$

$$D' = D + \delta\big(|U \cap L| - |V \cap L|\big)$$

$$= D + \delta\Big(\underbrace{|U \cap L| + |U \setminus L|}_{|U| = n/2} - \big(\underbrace{|U \setminus L| + |V \cap L|}_{\text{vertex cover of size } |M|}\big)\Big) = D + \delta\left(\frac{n}{2} - |M|\right)$$

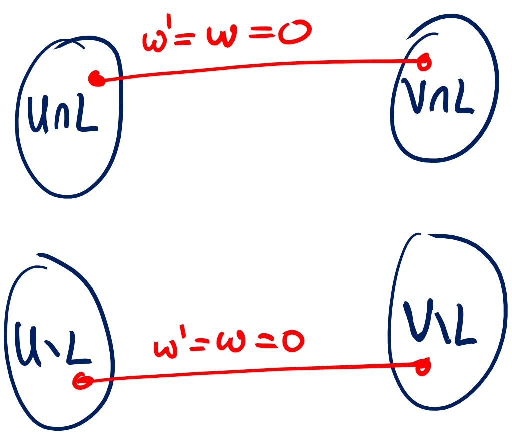
Termination

Some terminology

L'

- Old dual solution: $a_u,\quad b_v,\quad w_{uv} := c_{uv} - a_u - b_v$
- New dual solution: $a'_u,\quad b'_v,\quad w'_{uv} := c_{uv} - a'_u - b'_v$
- $E_0 := \{(u,v) : w_{uv} = 0\},\quad E'_0 := \{(u,v) : w'_{uv} = 0\}$
- M, M' : max. cardinality matchings of graphs ind. By $E_0,\ E'_0$

Claim: $|\boldsymbol{M'}| \geq |\boldsymbol{M}|$ and if $|M'| = |M|$, we can assume that $M = M'$.



Termination

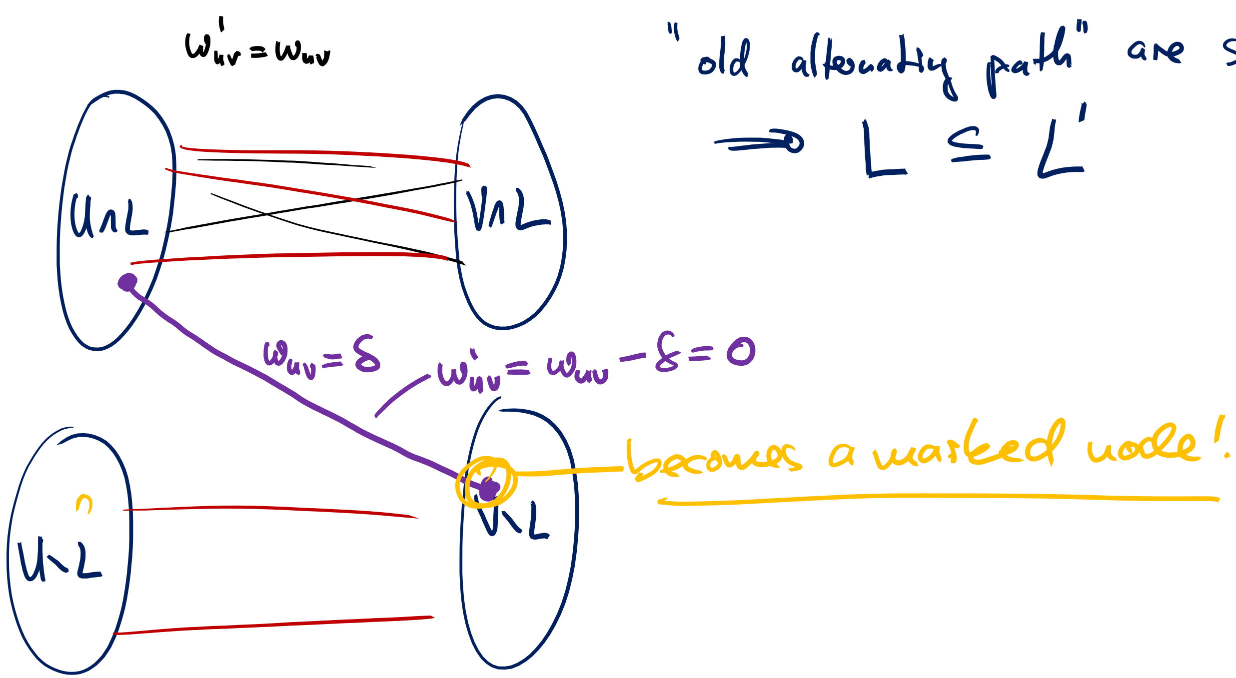
Lemma: The algorithm terminates in at most $O(n^2)$ iterations.

Proof:

- Each iteration: $M' > M$ or $M' = M$ and $|V \cap L'| > |V \cap L|$

"old alternating path" are still alt. paths

$\Rightarrow L \subseteq L'$



Min. Weight Perfect Matching: Summary

Theorem: A minimum weight perfect matching can be computed in time $O(n^4)$.

- First dual solution: e.g., $a_u = 0,\ b_v = \min_{u \in U} c_{uv}$

  $a_u = 0,\ b_v = 0$ also works

- Compute set E_0: $O(n^2)$

  $O(n^2)$ edges

- Compute max. cardinality matching of graph induced by E_0
  - First iteration: $O(n^2) \cdot O(n) = O(n^3)$
  - Other iterations: $O(n^2) \cdot O(1 + |M'| - |M|)$

  total cost when improving matching: $O(n^3)$

  total cost when $|M'| = |M|$: $O(n^3)$ for each matching size

  marking: $O(n^2)$ (given E_0 & M)

Matching Algorithms

We have seen:

- $O(mn)$ time alg. to compute a max. matching in *bipartite graphs*
- $O(mn^2)$ time alg. to compute a max. matching in *general graphs*

Better algorithms:

- Best known running time (bipartite and general gr.): $O(m\sqrt{n})$

Weighted matching:

- Edges have weight, find a matching of **maximum total weight**
- *Bipartite graphs*: polynomial-time primal-dual algorithm
- *General graphs*: can also be solved in polynomial time
  (Edmond's algorithms is used as blackbox)