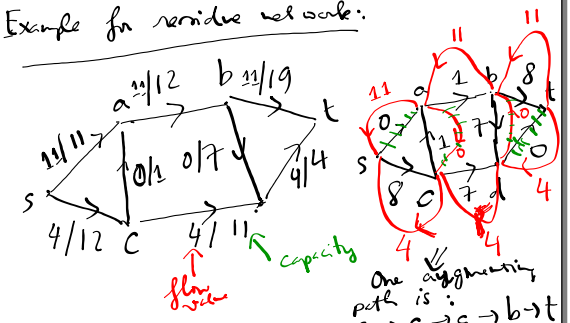
24 April 2023
MAT 2250

Example for residue network:



So we can use Algorithm in § 7.2 to find augmenting paths in ~~Breadth~~ the residual network.

Depending on the algorithm to find augmenting paths, we have different versions of the Ford & Fulkerson algorithm for maximal flow problems.

Edmonds - Karp algorithm uses Breadth First search.
Example on how the Breadth first search & Depth first search illustrated on the example in Wikipedia page

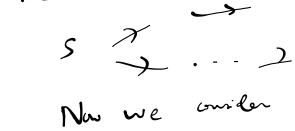
for Ford - Fulkerson algorithm.

Edmonds - Karp algorithm has the good property that it terminates even for non - integer networks. See Wikipedia an Ford - Fulkerson ~~not~~ algorithm for an example when Ford - Fulkerson does not terminate.

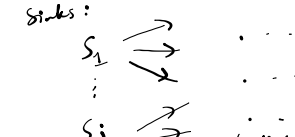
Some applications:
(See Wikipedia page on Circulation - problem)

Multiple sources & sinks:

So far we have only 1 source & 1 sink:



Now we consider many sources & many sinks:



We want a flow such that:
* If x is not a source or sink $\text{df}(x) = 0$

* Want the value

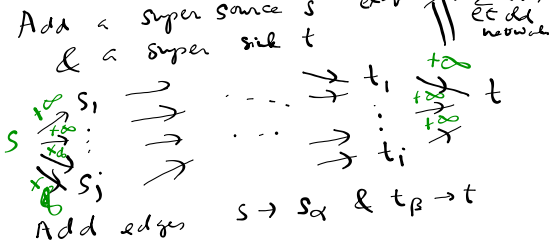
$$w(f) = \sum_{\alpha=1}^{j} \partial f(y_\alpha)$$

is maximal.

If you don't like $+\infty$, just replace it by a large number for example $\sum_{e \in E} c(e)$ of the network

How to solve this:

Add a super source s & a super sink t



Add edges $s \to s_\alpha$ & $t_\beta \to t$

The capacities on the new edges is $+\infty$ (infinity)

Now in the new network we have only 1 source s & 1 sink t. So we reduce to what we knew.

Now $\partial f(s_\alpha) = 0 \quad \forall \alpha, \beta$

$\partial f(t_\beta) = 0$

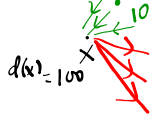
$\Rightarrow \sum_{\beta=1}^{i} \partial f(t_\beta) = \partial f(t)$

So maximal flow for the new network is the same as maximal flow for the old network.

Supply - demand problem:

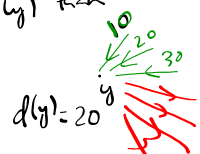
This is a multiple sources & sinks problem, with extra constraints:

A source vertex x is a supply node with supply capacity $d(x)$: meaning that x can flow out $d(x)$ more than what is flow in x:



$\Rightarrow$ # red is allowed to be $\leq (7+8+10)$ flow in $+ \underbrace{100}_{\text{supply capacity}} = 125$

A demand node / sink vertex y is a demand node with demand $d(y)$, meaning that it needs to flow at out least $d(y)$ than what is flow in it.

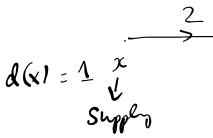


$\Rightarrow$ # red is $\geq (10+20+30) + 20 = 80$

The other vertex stisfy the usual condition $\partial f(v) = 0$, $v \neq x_i, v \neq y_j$

Rmk: There may not be exist any flow which satisfies the supply - demand problem.

Example:



x can only flow out at most 1 because $d(x) = 1$. But so y can receive at most 1. But y wants to receive at least 2, because $d(y) = 2$.

So this supply-demand problem is not an optimization problem, but a feasibility problem: Is there an object which satisfies the given constraints?

Idea:

Add super source s

super sink t ← supply nodes

Add Edges $s \to x_i$ with capacity $= d(x_i)$

Add Edges $y_j \to t$, with capacity $= d(y_j)$ (or probably, because ∞ is better)

demand node

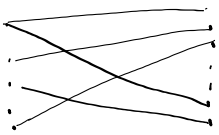


Theorem: An admissible flow f in G corresponds to an admissible flow f' in G' such that

$$f'(y_j \to t) \geq d(y_j) \quad \forall j.$$

Proof: Similar to what we proved for the reducing of multiple sources & sinks to super source & super sink. Explain with the picture from previous slide. (Q.E.D.)

Maximal matching on bipartite graph as a maximal flow problem

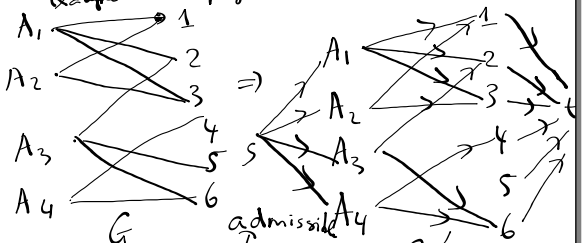


Want to find a matching with maximal # of edges. We do as follows: Construct a new directed graph G':

Vertices $= S \cup T \cup \{s\} \cup \{t\}$

Edges $:=$ { old edges in G, but with source, sink only 1 direction from S to T.
Edges $s \to u$, $u \in S$
$v \to t$, $v \in T$.

Capacity: capacity of every edge = 1.
Example on page 156 in the book



Capacity of every edge in G' is 1 so every flow will have value 0 or 1 on each edge.
$\partial f(A_i) = 0$ means that $\#\{j : f(A_i, j) = 1\} \leq 1$.

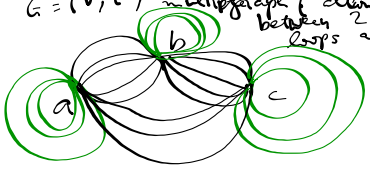
So an admissible flow for G' corresponds to a matching in G.

Rmk: The circulation problem can be reduced to the supply-demand problem.

---

§ 8.4 Euler graphs & the Travelling ~~seller~~ Salesman Problem (TSP)

$G = (V, E)$ multigraph (allow many edges between 2 vertices, also loops at 1 vertex)

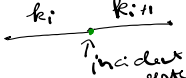


Euler circuit:

An edge sequence

$k_1, k_2, \dots, k_q$

where k_i & k_{i+1} are incident (i.e the end point of k_i is the starting vertex of k_{i+1}):



such that $k_i \neq k_j$ for $i \neq j$
& $q = |E|$
& the starting point of k_1 is the end point of k_q.

Rmk: If we don't require that the starting point of k_1 is the end point of k_q, then we have an open Euler circuit $= (V, E)$

Recall: An edge k in G is a cut-edge if $(V, E \setminus \{k\})$ is not connected.



Algorithm 8.20: Find an Euler circuit of a connected-graph $G = (V, E)$.

Step 0: * Start with a vertex u_0

Step i: * Assume that we already have $\{u_0, \dots, u_i\}$ & edges $k_1 = u_0 \to u_1, \dots, k_i = u_{i-1} \to u_i$

Then we add an edge $u_i \to u_{i+1}$ if there is such an edge, with the following

Caution: if $\exists$ an edge $u_i \to u_j$ not a cutting edge then we add it, otherwise we can add a cutting edge.

in $G_i = (V, E \setminus \{k_1, \dots, k_i\})$