Edge-disjoint Paths

Given directed $G=(V,E)$ + $s, t \in V$.

Goal: find max size collection of edge disjoint (s,t)-paths.
(no two paths share an edge)

Alg: Set capacities of all edges to 1...

If there are k edge-disjoint paths, we can make a flow f of value k by setting $f(e)=1$ iff e is in a path.

So compute max flow f^*.

$f^*(e)$ is an integer $\forall e$.
$\Rightarrow f^*(e) \in \{0,1\} \; \forall e$.

$E' \subseteq E$: Set of edges e s.t. $f^*(e)=1$.

Find an (s,t)-path P in E'.
Decrease $f^*(e)$ by 1 on all edges of P.

Inductively, we can repeat process untill we have $|f^*|-1$ paths in addition to P

So we found $|f^*|$ paths!

Time: Use Orlin for $O(VE)$ time.

-or- if G is simple, (no parallel edges)

$$|f^*| \leq |\{s\}, V \setminus \{s\}|$$
$$\leq |V| - 1$$

Use Ford-Fulkerson in

$$O(E|f^*|) = O(VE) \text{ time.}$$

Vertex Capacities (or both)

Have $c : V \to \mathbb{R}_{\geq 0}$.

Flow f is feasible if

$$\sum_{u \to v} f(u \to v) \leq c(v).$$

Reduction to only edge capacities:

Replace each vertex v with v_{in} + v_{out}.

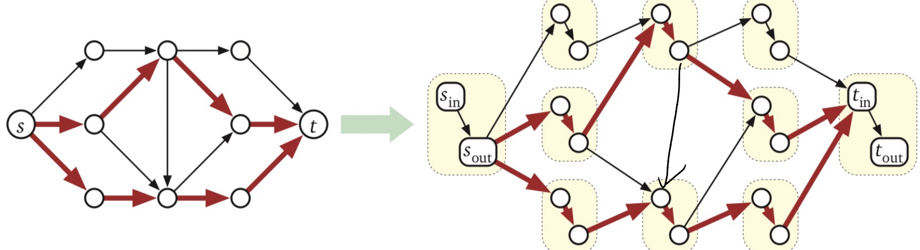
Add edges $v_{in} \to v_{out}$ of capacity $c(v_{in} \to v_{out}) := c(v)$.

Each edge $u \to v$ of input graph becomes $u_{out} \to v_{in}$.

Feasible flows in one graph are feasible in other: just copy flow values between original edges.

Can reduce vertex + edge capacities to edge only by setting $c(u_{out} \to v_{in}) := c(u \to v)$ also.

Or do vertex-disjoint paths!

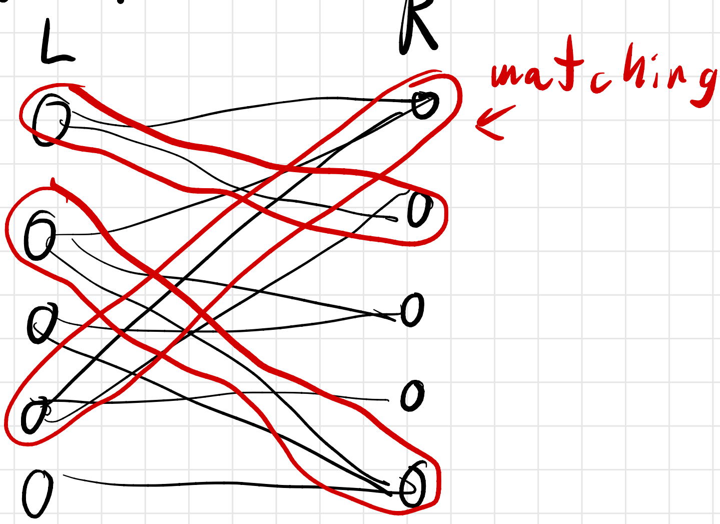


Bipartite Matching

Given undirected bipartite $G = (L \cup R, E)$.

($L \cap R = \emptyset$, every edge goes from L to R)

A <u>matching</u> is a subgraph where all vertices have degree at most 1.

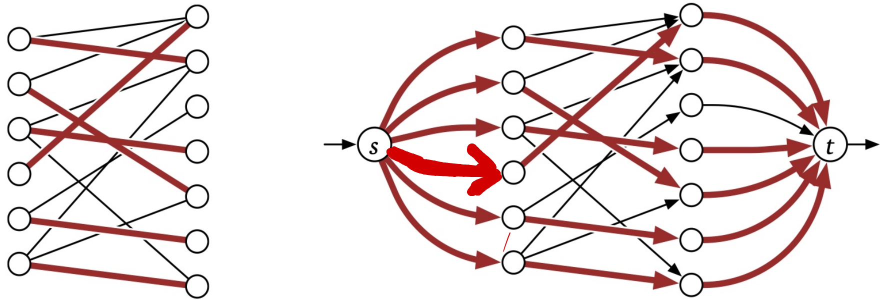


Goal: Compute max size matching.

Reduction to max flow:

Make directed G' by taking G and...

- orient edges from L to R
- add vertices s + t
- add edge $s \to \ell$ $\forall$ $\ell \in L$
- add edge $r \to t$ $\forall$ $r \in R$
- assign capacity 1 to all edges



Take any matching M...

make a flow f_m:

$\forall\ lr \in M$, send one unit of flow along $s \to l \to r \to t$

$|f_m| = |M| \Rightarrow |f^*| \geq$ size of max matching

Take any integer max flow f^*.

Sends ≤ 1 units out of any $l \in L$

in any $r \in R$

$M :=$ edges $l \to r$ with flow

$|M| = |f^*|$

$|f^*| \leq V/2 \Rightarrow$ Ford-Fulkerson takes $O(VE)$ time.

Exam Scheduling

- n classes,
- r rooms,
- t time slots, and
- p proctors

- at most one class's exam in each room in each time slot.
- no splitting a class into multiple rooms or time slots
- a proctors
  - oversees ≤ 1 exam at a time
  - only available at certain times varying per proctor
  - oversees ≤ 5 exams total

- class has to fit in room!

Formal Input:

Integer array $E[1..n]$:

$E[i]$ students enrolled in class i

Integer array $S[1..r]$:

Room j has $S[j]$ seats.

(class i's final can be held in room j iff $E[i] \leq S[j]$)

Boolean array $A[1..t, 1..p]$:

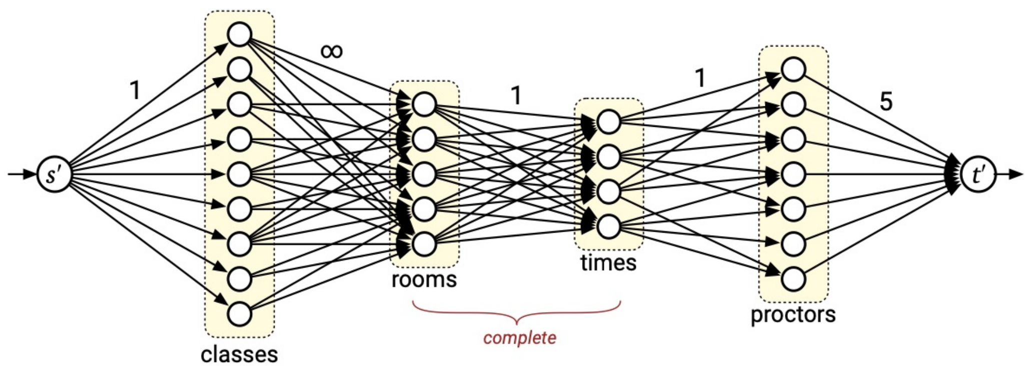
$A[k, l]$ = True iff proctor l is available during kth time slot

Construct graph G with six types of vertices

- source s'
- c_i for each class i
- r_j for each room j
- t_k for each time slot k
- p_l for each proctor l
- sink t'

and five types of edges

- edge $s \to c_i$ of cap. 1 $\forall i$
- edge $c_i \to r_j$ of cap. ∞ $\forall E[i] \in S[j]$
- edge $r_j \to t_k$ of cap. 1 $\forall (j,k)$
- edge $t_k \to p_\ell$ of cap. 1 $\forall A[k,\ell]=\text{True}$
- edge $p_\ell \to t'$ of cap. 5 $\forall \ell$



G has $n+r+t+p+2$ vertices & $O(nr+rt+tp)$ edges

Compute max (s', t')-flow!

Assignments lead to flow paths

Schedule all classes for a flow of value n.

$|f^*| \leq n$

Peel off paths for a good exam schedule.

① Can schedule all exams iff $|f^*| = n$.

$O(VE) = O((n+r+T+p)(nr+rT+Tp))$ time.