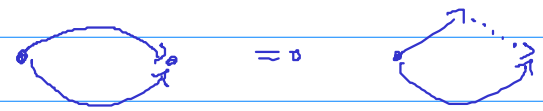
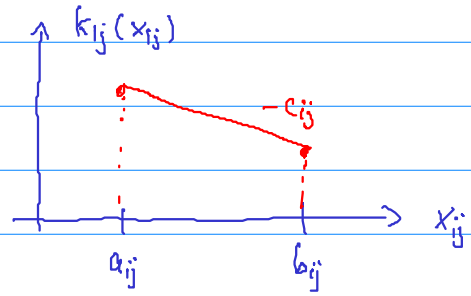
§19 Time cost tradeoff problems

The linear case

Given
- project network as arc diagram without parallel jobs



- for every job (i,j) an interval $I_{ij} = [a_{ij}, b_{ij}]$ of possible job durations
- for every job (i,j) a cost function k_{ij} with slope $-c_{ij}$, $c_{ij} \geq 0$



$k_{ij}(x_{ij})$ denotes the cost of processing (i,j) with processing time x_{ij}

- time limit t for the makespan

Goal: Execute the project at minimum cost within the time limit t

i.e. $$\min k(x) := \sum_{\text{jobs } (i,j)} k_{ij}(x_{ij}) =: H(t)$$

s.t. x = vector of chosen x_{ij}

and $C_{max}(x) \leq t$

$H(t)$ = min cost for t

$H(t)$ is called the project cost curve

This problem is called the (linear) time cost tradeoff problem (tcto)

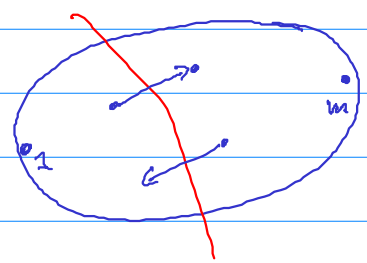
Note: $$k_{ij}(x_{ij}) = \underbrace{k_{ij}(b_{ij})}_{\text{constant}} + \underbrace{(b_{ij} - x_{ij})}_{\text{shortening from } b_{ij}} \cdot c_{ij}$$

$\Rightarrow$ we shorten jobs at the cost rate c_{ij}
and want to find the right shortenings

Basic idea:

- consider an optimal vector x for t
- characterize "optimal" tradeoffs to $t-\varepsilon$, ε small, in the arc diagram
  [will show: must shorten on a "good" cut in the network of critical jobs]

Def: Let $D = (N, A)$ be the arc diagram of G
$N := \{1, \dots, m\}$ $\quad 1 \hat{=}$ source, $m \hat{=}$ sink
A cut $[S, T]$ of D is a partition $N = S \dot{\cup} T$ of N
with $1 \in S$, $m \in T$
$m \notin S$ $\quad 1 \notin T$



forward arc (i,j) $\quad i \in S$, $j \in T$
backward arc (i,j) $\quad j \in S$, $i \in T$

(i,j) is <u>critical</u> if it is on a critical path (for a given x)
$D_{crit} = (N_{crit}, A_{crit})$ denotes the subnetwork of critical jobs

to test criticality for a given x of job (i,j) compute
length of longest path from 1 to i $\longrightarrow \pi_i$
length of longest path from j to m $\longrightarrow \sigma_j$
$\Rightarrow$ π_i = earliest start of (i,j)
$\pi_m - \sigma_j$ = latest finish of (i,j) to stay within the makespan π_m

(i,j) is critical $\Leftrightarrow$ $\pi_m - \sigma_j - \pi_i = x_{ij}$

Given x, let $\pi_i(x)$ denote the length of a longest path from 1 to i w.r.t. x

Let x be optimal for t $\longrightarrow$ node potentials $\pi_i(x)$

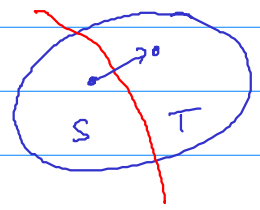
Let z .... $t-\varepsilon$ $\longrightarrow$ " $\pi_i(z)$

$\Rightarrow$ $\pi_1(x) = \pi_1(z) = 0$

$\pi_m(x) = t$, $\pi_m(z) = t-\varepsilon$

$\Rightarrow$ $S := \{ i \in N \mid \pi_i(x) = \pi_i(z) \}$ $T := N \setminus S$ is a cut

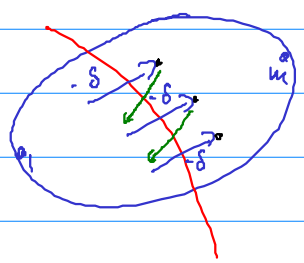
$\Downarrow$



processing times have changed on forward arcs of that cut (and maybe elsewhere)

$\Downarrow$

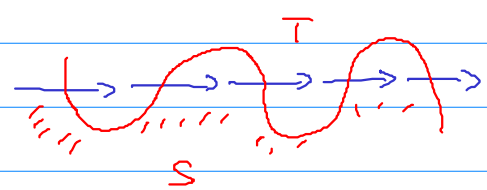
Idea: Consider what happens if we change proc times on a cut only



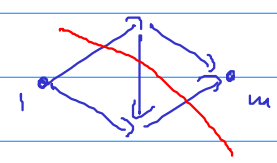
shorten forward arcs by δ
$\Rightarrow$ reduces π_m by at least δ (19.1)

But: some $\pi_i(z)$ may be changed by a multiple of δ

A cut $[S,T]$ is <u>good</u> if every $s \in S$ can be reached from 1 by a directed path in S



not a good cut

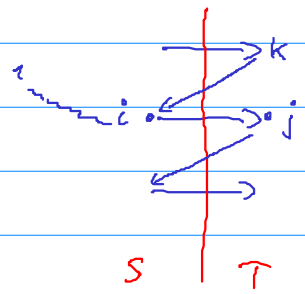


a good cut

Shortening all forward arcs by δ on a good cut reduces π_m by exactly δ } (19.2)

Proof of (19.2) $[S,T]$ good $\Rightarrow$ $\exists$ path from 1 to i in S

$\Rightarrow$ π_j is reduced by exactly δ

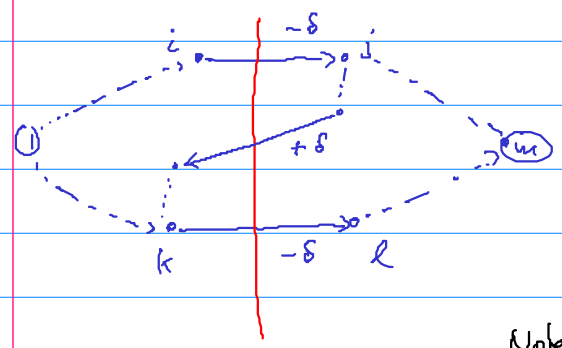


$\Rightarrow$ all start nodes of paths in T are shortened by exactly δ

But now backward arcs (k,i) have slack δ (i.e. $\pi_m - b_i - \pi_k > x_{ki}$)
i.e. they are no longer critical
This can be exploited by lengthening backward arcs (if possible)

Let (S,T) be a good cut
shorten all forward arcs by δ
lengthen all backward arcs by δ } $\Rightarrow$ π_m is shortened by δ (19.3)

Proof of (19.3)



Show that if (i,j) is a forward arc, then π_j is reduced by at least δ
The arguments from (19.2) remain valid
$\Rightarrow$ π_m is reduced by exactly δ

Note: every backward arc is contained between two forward arcs

Def: Shortening on a good cut:
shorten all forward arcs by δ (if their processing time permits, i.e. $x_{ij} > a_{ij}$
lengthen all backward arcs by δ (..... $x_{ij} < b_{ij}$)

<u>cost rate of a [good] cut</u> [cost per unit of decrease]

$$\text{costrate}[S,T] = \sum_{\substack{(i,j) \\ \text{forward arc} \\ (x_{ij} > a_{ij})}} c_{ij} - \sum_{\substack{(k,\ell) \\ k,\ell \text{ backward arc} \\ x_{k\ell} < b_{k\ell}}} c_{k\ell}$$

<u>How big can δ be ?</u>

$$\delta = \min \{ \delta_1, \delta_2, \delta_3 \}$$

δ_1 = amount of decrease until a non-critical job becomes critical

$$= \min \{ \underbrace{\pi_m - \sigma_j - \pi_i - x_{ij}}_{\text{slack of } (i,j)} \mid (i,j) \text{ not critical} \}$$

δ_2 = amount of decrease until a forward arc is at its minimum processing time

$$= \min \{ x_{ij} - a_{ij} \mid (i,j) \text{ forward arc in the cut} \}$$

δ_3 = amount of increase until a backward arc that can be prolonged reaches its maximum processing time

$$= \min \{ b_{ij} - x_{ij} \mid (i,j) \text{ backward arc and } x_{ij} < b_{ij} \}$$

19.1 THEOREM: Let x be optimal for $t > C_{max}(a)$ [= min. makespan]

Then there exists a good cut $[S,T]$ in D_{crit} with ass. $\delta > 0$ s.t.

for every $\rho \in]0, \delta]$, the change of processing times according to (19.3)

by ρ yields an optimal processing time vector y^ρ for $t-\rho$.

So

$$y^\rho_{ij} = \begin{cases} x_{ij} - \rho & (i,j) \text{ is a forward arc of } [S,T] \\ x_{ij} + \rho & (i,j) \text{ is a backward arc of } [S,T] \text{ with } x_{ij} < b_{ij} \\ x_{ij} & \text{otherwise,} \end{cases}$$

The total cost grows by the amount $\rho \cdot \text{cost rate}(S,T)$

[Proof later]

19.2 COROLLARY: The project cost curve $H(t)$ is piecewise linear and convex on $[t_{min}, t_{max}]$ with

$$t_{min} = C_{max}(a_1, \dots, a_n) \quad \leftarrow \text{ min duration per job}$$
$$t_{max} = C_{max}(b_1 \dots b_n) \quad \leftarrow \text{ max duration per job}$$

It may be constructed as follows:

(1) Start at $t = t_{max}$. Then $x = (b_1 \dots b_n)$ is optimal

(2) Repeat until $t = t_{min}$

(a) Construct D_{crit} with respect to x

(b) Find a good cut with minimum cost rate that can still be shortened

(c) Compute δ of this cut and change the processing times according to (19.3)

(d) If $t - \delta \leq t_{min}$, set $t = t_{min}$

Else set $t := t - \delta$ and let x be the new optimal vector for $t - \delta$ according to (19.3)

So linear pieces of $H(t)$ correspond to (one or more) cuts with cost rate equal to the slope of that piece

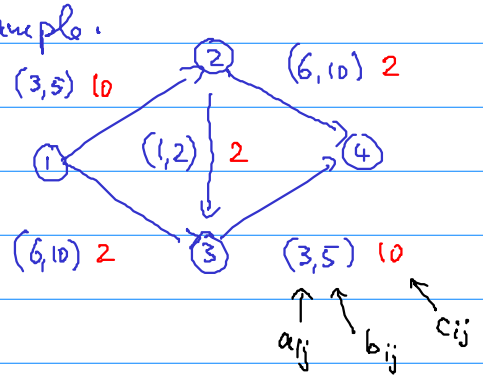
Proof. Thm 19.1 and the integrality of the a_{ij}, b_{ij} $(\Rightarrow \delta \geq 1)$

$\Downarrow$

termination

convexity $\rightarrow$ later $\square$

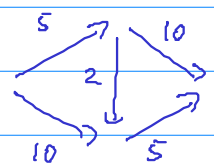
Example.



$\Rightarrow t_{max} = 15$

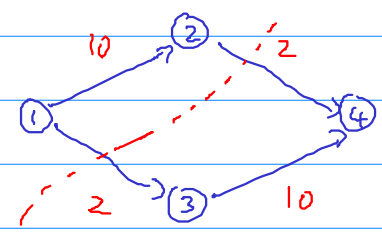
$t_{min} = 9$

all jobs at maximum duration



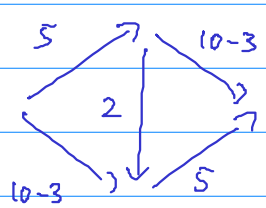
$C_{max} = 15$

D_{crit}



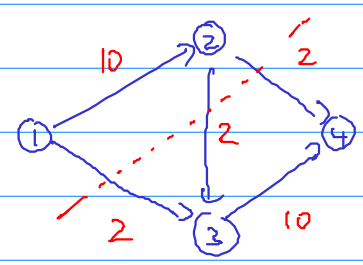
cost rate 4

$\left.\begin{array}{l} \delta_1 = 3 \\ \delta_2 = 4 \\ \delta_3 = \infty \end{array}\right\} \Rightarrow \delta = 3$



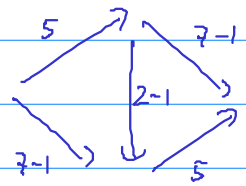
$C_{max} = 12$

D_{crit}



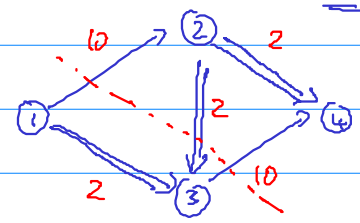
cost rate $= 6$

$\delta = 1$



$C_{max} = 11$

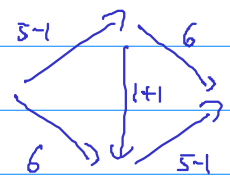
D_{crit}



$\Longrightarrow$ $\hat{=}$ cannot be shortened

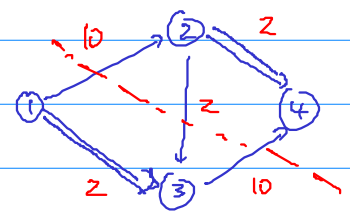
cost rate = 18

$\delta_2 = 2 \quad \delta_3 = 1 \quad \delta_1 = \infty$



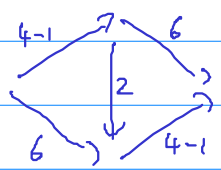
$C_{max} = 10$

D_{crit}



cost rate = 20

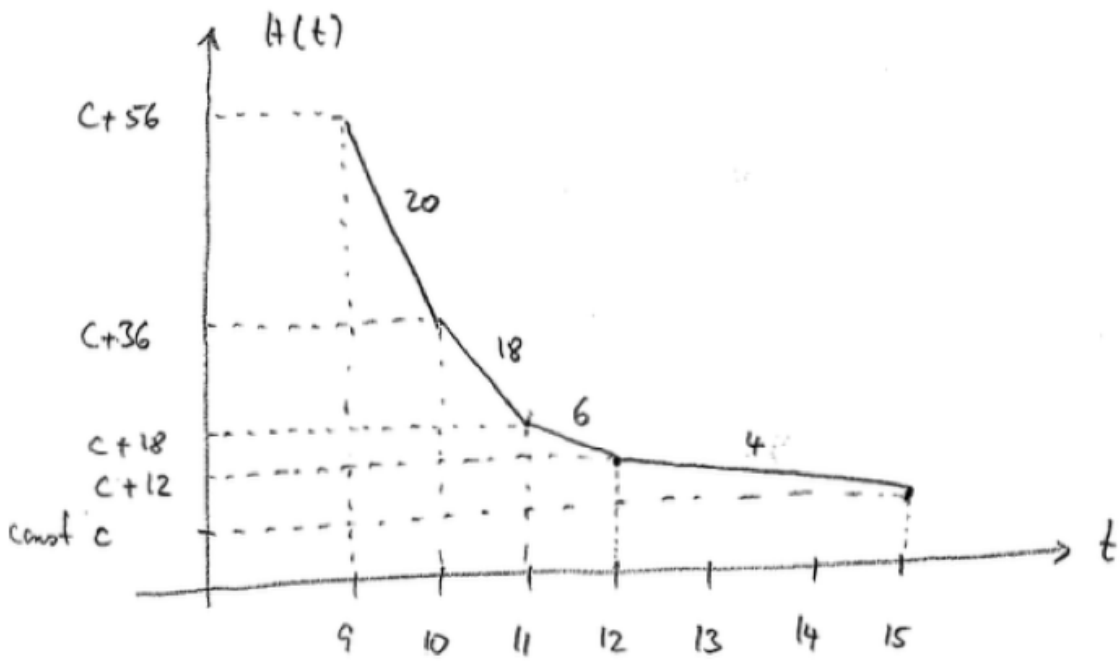
$\delta_2 = 1$



$C_{max} = 9$

$\Rightarrow$ t_{min} reached

Note that after shortening, job (2,3) is no longer critical



Question: How to find the good cuts algorithmically?

Idea: Use network flows

Max flow - min cut theorem:

the maximum value of an 1-m-flow = min capacity of a 1,m-cut

feasible 1,m-flow: flow conversation at every node $i \neq 1, m$

capacities respected $\quad l_{ij} \leq x_{ij} \leq u_{ij}$

lower capacity

upper capacity

flow value on arc (i,j)

value of a flow = net outflow out of source 1

$$= \sum_{\substack{j \\ (1,j)\text{ arc}}} x_{1j} - \sum_{\substack{j \\ (j,1)\text{ arc}}} x_{j1}$$

capacity of cut $[S,T]$

$$\text{cap}(S,T) = \sum_{(i,j)\text{ forward arc}} u_{ij} - \sum_{(t,s)\text{ backward arc}} l_{t,s}$$

$$\text{costrate}(S,T) = \sum_{(i,j)\text{ forward arc}} c_{ij} - \sum_{(t,s)\text{ backward arc}} c_{ts}$$

$x_{ij} > a_{ij} \qquad x_{ts} < b_{ts}$

So, in the current network D_{crit}, put

$$u_{ij} := \begin{cases} c_{ij} & \text{if } x_{ij} > a_{ij} \\ \infty & \text{otherwise} \end{cases}$$

[(i,j) can be shortened]

[$\Rightarrow$ do not want such an arc in a cut]

$$l_{ij} := \begin{cases} c_{ij} & \text{if } x_{ij} < b_{ij} \\ 0 & \text{otherwise} \end{cases}$$

[(i,j) can be lengthened]

[will not contribute to cost rate]

Then (from flow theory):

- any max flow algorithm producing good cuts will find a maximum flow and a good cut of minimum capacity = min cost rate
- HERE: find flow augmenting paths by BFS
  BFS guaranties (when there is no augmenting path) that the cut is good

19.4 THEOREM:

(1) Every cut with minimum cost rate can be found in

$$O\left(\#\text{nodes} \cdot \#\text{arcs} \cdot \log\left(\frac{(\#\text{nodes})^2}{\#\text{arcs}}\right)\right) = O(n^3 \log n)$$

Goldberg & Tarjan &

(2) The zero-flow is feasible at t_{max}.

The current flow remains feasible when the capacities are changed

$\Rightarrow H(t)$ convex!

(3) $H(t)$ can be calculated in $O(\underbrace{\#\text{cuts calculated}}_{\geq \#\text{breakpoints}} \cdot n^3 \log n)$

may be exponential in general

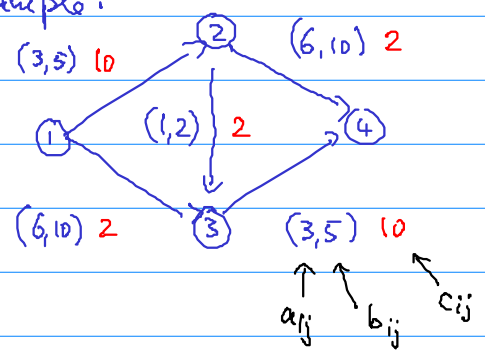
Proof: (1) flow theory

(2) easy verification $\Rightarrow$ cost rates are increasing $\Rightarrow$ $H(t)$ convex

(3) obvious, that there may be exponentially many breakpoints

uses a (non-trivial) example

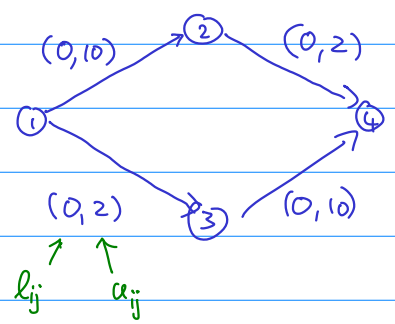
Example.



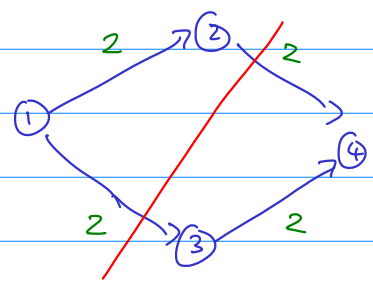
$\Rightarrow$ $t_{max} = 15$

$t_{min} = 9$

1st flow network

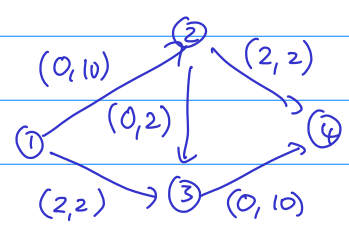


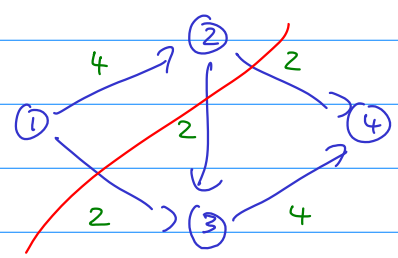
max flow and min cut



$v = 4$

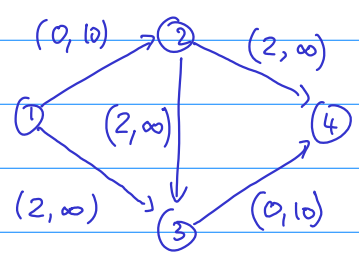
2nd flow network

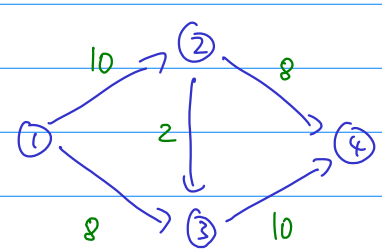




$v = 6$

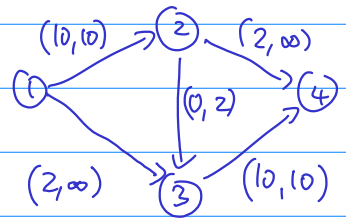
3rd flow network

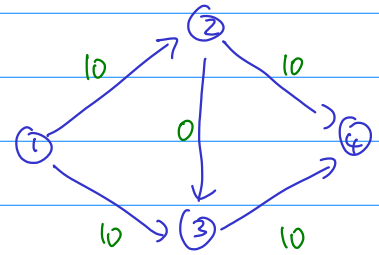




$v = 18$

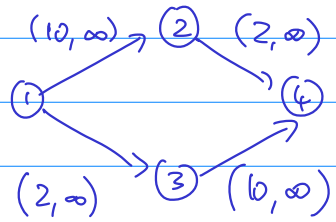
4th flow network

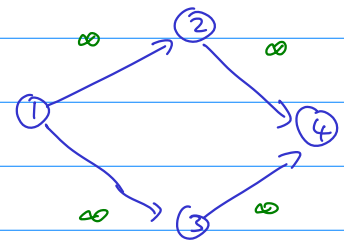




$v = 20$

5th flow network





$\infty \Rightarrow t_{min}$ reached

It remains to prove Theorem 19.1

19.1 THEOREM: Let x be optimal for $t > C_{max}(a)$ [= min. makespan]

Then there exists a good cut $[S,T]$ in D_{crit} with ass. $\rho > 0$ s.t.

for every $\rho \in]0, \delta]$, the change of processing times according to (19.3)

by ρ yields an optimal processing time vector y^ρ for $t-\rho$.

So

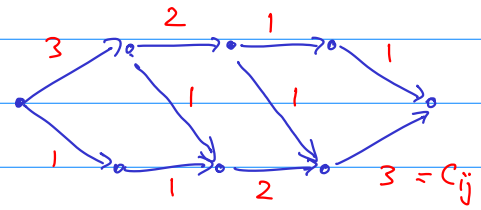
$$y^\rho_{ij} = \begin{cases} x_{ij} - \rho & (i,j) \text{ is a forward arc of } [S,T] \\ x_{ij} + \rho & (i,j) \text{ is a backward arc of } [S,T] \text{ with } x_{ij} < b_{ij} \\ x_{ij} & \text{otherwise,} \end{cases}$$

The total cost grows by the amount $\rho \cdot \text{cost rate}(S,T)$

[Proof later]

Proof (sketch on an example)

Expl:



every job has $a_{ij} = 1$, $b_{ij} = 10$

Consider $t \in [t_{min}, t_{max}]$ and x optimal for t

Problem: an optimal vector z for $t-g$ may obtained by changing several jobs in D_{crit} (not only on a cut)

Must show: this can be done at the same cost on a good cut

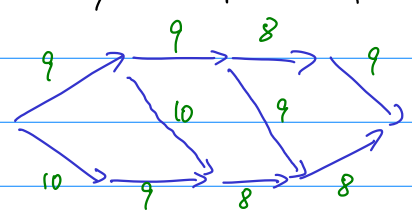
Expl: $x = (10, 10 \dots 10)$ is optimal for $t = t_{max} = 40$ and every job is critical

Consider $\bar{\delta}$ defined as δ , but over all arcs of D_{crit}
($\bar{\delta} \leq \delta \leftarrow$ calculated for a good cut)

Expl:
$$\left.\begin{array}{lll} \bar{\delta}_1 = \infty & & \text{(all arcs are critical)} \\ \bar{\delta}_2 = 9 & & \text{(all arcs can be shortened by 9)} \\ \bar{\delta}_3 = \infty & & \text{(no arc can be lengthened)} \end{array}\right\} \Rightarrow \bar{\delta} = 9$$

Let $g_0 \in]0, \bar{\delta}]$ and let y be optimal for $t - g_0$

Expl: $g_0 = 5$ $\quad y$

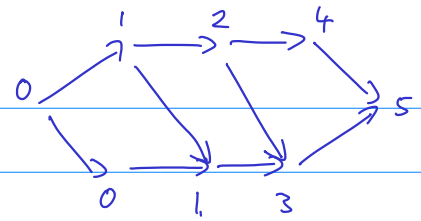


$t - g_0 = 35$

Define for $i \in N$ the potential difference $\delta\pi_i = \pi_i(x) - \pi_i(y)$

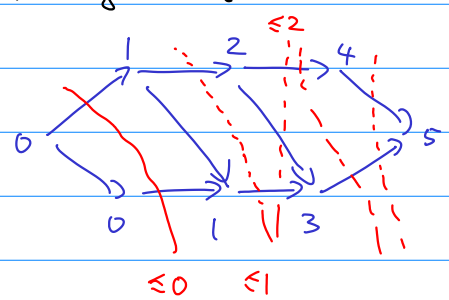
Let $\Delta\pi_1 < \Delta\pi_2 < \Delta\pi_3 < \dots < \Delta\pi_\ell$ be the different $\delta\pi_i$ values

Expl:



$\Delta\pi_1$	$\Delta\pi_2$	$\Delta\pi_3$	$\Delta\pi_4$	$\Delta\pi_5$	$\Delta\pi_6$
0	1	2	3	4	5

(1) $[S_k, T_k]$ with $S_k := \{ i \in N \mid \delta\pi_i \leq \Delta\pi_k \}$ $\quad T_k = N \setminus S_k$
is a good cut $\forall$ k



(2) y is obtained from x as follows
for k := 1 to $\ell - 1$ do
(i) $\Delta_k := \Delta\pi_k - \Delta\pi_{k-1}$ $\quad (\Delta\pi_0 := 0)$
(ii) $x_{ij} := x_{ij} - \Delta_k$ for all forward arcs in $[S_k, T_k]$
(iii) $x_{ij} := x_{ij} + \Delta_k$ " backward " "

Expl: $\Delta_k = 1$, no backward arcs
$\Rightarrow$ subtract 1 on every forward arc of every cut

(3) Let $[S_r, T_r]$ be the cut with smallest cost rate among these cuts
Let z be the proc. time vector obtained from x on $[S_r, T_r]$ by ϱ_0
(possible $\varrho_0 \leq \bar{\delta}$)
Then z optimal for $t - \varrho_0$

Proof: change of total cost for $x \to y$:

$$\sum_{k=1}^{\ell-1} \text{costrate } [S_k, T_k] \cdot \Delta_k \geq \text{costrate } [S_r, T_r] \cdot \varrho_0$$

cost for $x \to z$

(4) Decrease on cut $[S_r, T_r]$ is optimal for <u>all</u> $\varrho_0 \in [0, \bar{\delta}]$
(ϱ_0 was fixed so far)

Let $\varrho_1, \varrho_2 \in [0, \bar{\delta}]$ with best cuts $[\bar{S}_1, \bar{T}_1]$, $[\bar{S}_2, \bar{T}_2]$

assume w.l.o.g. that $\varrho_1 \leq \varrho_2$

Show cost rate $[\bar{S}_1, \bar{T}_1]$ = cost rate $[\bar{S}_2, \bar{T}_2]$

If "<" then decrease on $[\bar{S}_1, \bar{S}_2]$ by $\min\{\varrho_1, \varrho_2\}$ gives a better solution for $t - \varrho_1$ on the other cut, a contradiction

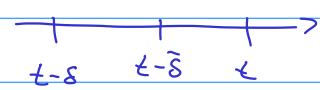
(4) So far decrease by at most $\bar{\delta}$

Now by at most δ (defined by the best cut $[S_r, T_r] =: [S, T]$

$\bar{\delta} < \delta \overset{(4)}{\Rightarrow} [S, T]$ is optimal for $t - \bar{\delta}$ with proc time vector z

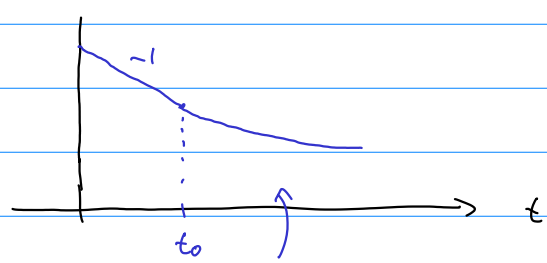
All cuts that can be decreased at time $t - \bar{\delta}$ and z can also be decreased at time t and x (by definition of $\bar{\delta}$)

$\Rightarrow [S, T]$ is still the best cut at $t - \bar{\delta}$ and now can be decreased further until $t - \delta$ $\square$



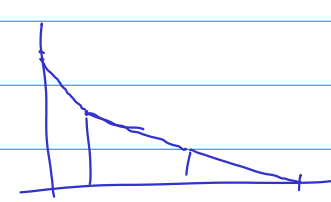
Back to the computation of

$$\psi(t) = \min \sum \mathbb{E}[X_{ij} - x_{ij}]^+ \qquad (*)$$
$$\text{s.t. } C_{max}(x) \leq t$$



Here $(*)$ is valid

piecewise linear and convex for discrete distributions



interpret as cost function of job (i,j)

$\Downarrow$

tcto problem with piecewise linear and convex cost functions

Exercises

19.1 Generalize the computation of $H(t)$ by flow methods to the case that the cost functions $k_{ij}(x_{ij})$ of every job (i,j) are piecewise linear and convex.