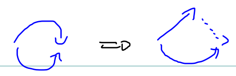
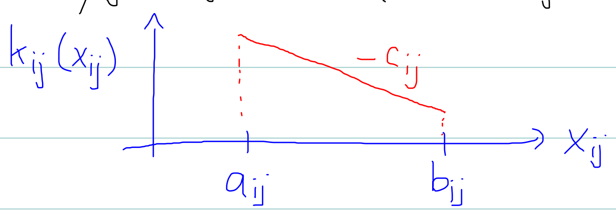
§19 Time-cost tradeoff problems

The linear case

Given:
- project network as arc diagram without parallel arcs 
  we denote a job by its pair (i,j) of nodes
- 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 $-c_{ij}$, $c_{ij} > 0$



  $k_{ij}(x_{ij})$ denotes the cost for processing job (i,j) with processing time (= duration) x_{ij}. x_{ij} can be chosen in $[a_{ij}, b_{ij}]$
- a time limit t for the makespan C_{max}

Goal: Execute the project at minimum cost within the given time limit, i.e.,

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

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

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

$H(t) := \min \{ k(x) \mid x_{ij} \in I_{ij},\ C_{max}(x) \le t \}$ as a function of t is called the project cost curve

The problem is called the (linear) time-cost tradeoff problem (tcto for short)

Note: $k_{ij}(x_{ij}) = \underbrace{k_{ij}(b_{ij})}_{\text{constant}} + \underbrace{(b_{ij} - x_{ij})}_{\text{shortening from } b_{ij}} c_{ij}$ $\Big\} \Rightarrow$ we shorten jobs at a cost rate c_{ij} and want to find the right shortenings.

Basic Idea:

- Consider an optimal proc. time vector x for t

  [Exists since k is continuous on $\underbrace{X\,[a_{ij}, b_{ij}]}_{\text{compact}}$ $\Rightarrow$ k attains the minimum]

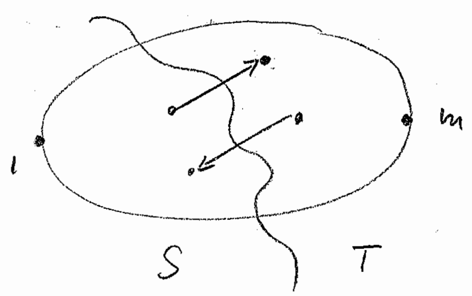
- Characterize "optimal" tradeoffs to $t-\varepsilon$, ε small, in the arc diagram

  [will show: must shorten on cut in the network of critical paths]

Def: Let $D = (N, A)$ be the arc diagram of G. $N = \{1 \dots m\}$

$1 \hat{=}$ source, m = sink

A cut $[S,T]$ of D is a partition $N = S \dot{\cup} T$ of N with $1 \in S$, $m \in$



forward arcs (i,j) : $i \in S$, $j \in T$

backward arcs (k,ℓ) : $k \in T$, $\ell \in S$

(i,j) is called critical if it is on a critical path (for given x)

$D_{crit} = (N_{crit}, E_{crit})$ denotes the subnetwork of critical paths.

Let, w.r.t. processing time vector x, $\pi_i(x)$ denote the length of a longest path from 1 to i

"potential" of node i

let x be optimal for t $\rightarrow$ node potentials $\pi_i(x)$

z be optimal for $t-\varepsilon$ $\rightarrow$ node potentials $\pi_i(z)$

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

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

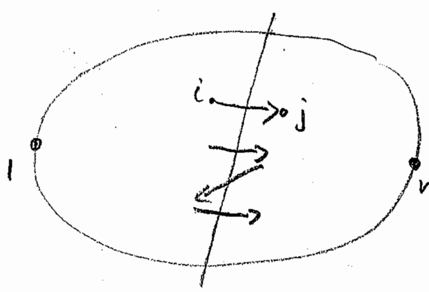
$\Rightarrow$ $S := \{ i \in N \mid \pi_i(x) = \pi_i(z) \}$, $T := N \setminus S$ is a cut

$\Rightarrow$ processing times have been changed on arcs of the cut

[and maybe elsewhere]

$\Downarrow$

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



shorten all forward arcs by δ

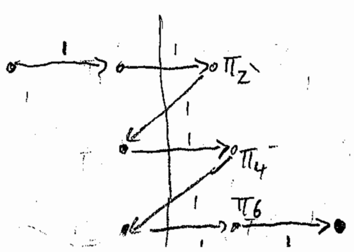
reduces π_m by at least δ $\Bigg\}$ (19.1)

Proof of (19.1)

(i,j) forward arc $\Rightarrow$ π_j is reduced by at least δ

Every critical path has a forward arc $\Rightarrow$ π_m reduced by at least δ

Some π_j may be shortened by a multiple of δ $\square$

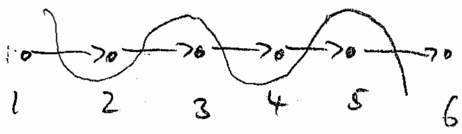


$\Rightarrow$ π_4 reduced by 2δ

π_6 reduced by 3δ

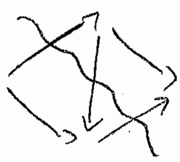
with right choice of x_{ij}

A cut $[S,T]$ is a <u>good cut</u> of D_{crit} if every $i \in S$ can be reached from 1 by a directed path in S.



not a good cut

$[\{1\}, \{2 \ldots 6\}]$ is good

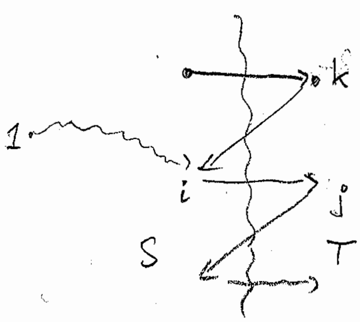


a good cut

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

Proof of (19.2):

let (i,j) be a forward arc of $[S,T]$



$[S,T]$ good $\Rightarrow$ There is a path from 1 to i in S

$\Rightarrow$ π_j is reduced exactly by δ

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

But now backward arcs (k,i) have slack δ

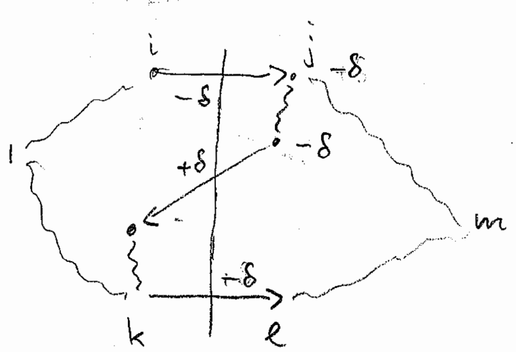
This can be exploited by lengthening backward arcs (if possible)

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

<u>Proof of (19.3)</u>

show that if (i, j) is a forward arc, then π_j is shortened by at least δ. The argument from (19.2) remains true

$\Rightarrow$ π_m is decreased by δ



Let (i,j) (k,ℓ) be forward arcs on a path with a backward arc between them

$\Rightarrow$ $-2\delta + \delta = -\delta$

every backward arc is contained between 2 forward arcs in this way $\square$

<u>Def</u> shortening on a [good] cut:

shorten all forward arcs by δ

lengthen all backward arcs by δ (if their processing time permits i.e. $x_{ij} < b_{ij}$)

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

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

<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

$= \min \{ \pi_j - \pi_i - x_{ij} \mid (i,j) \text{ not critical} \}$

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

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

Necessary is of course that <u>all forward arcs can be shortened</u>.

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

$t_{min} = C_{max}(a_1, \dots, a_n)$ ← min duration per job

$t_{max} = C_{max}(b_1 \dots b_n)$ ← 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 $t_1 = t_{min}$

  (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 $t = t_{min}$

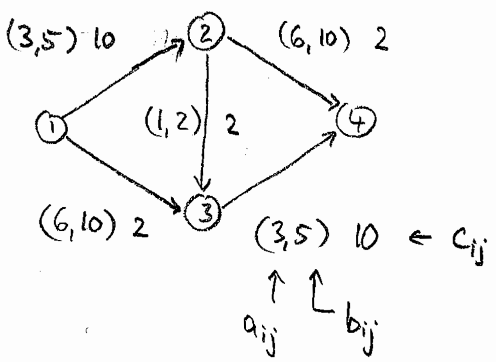
Else set $t := t - \delta$ and let x be the new optimal vector for $t - \delta$ according to (19.3)

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} ($\Rightarrow \delta \geq 1$) ↳ termination

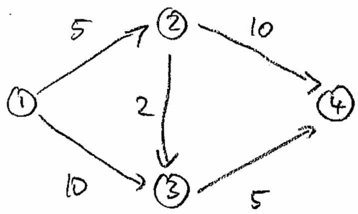
convexity → later □

19.3 Example:



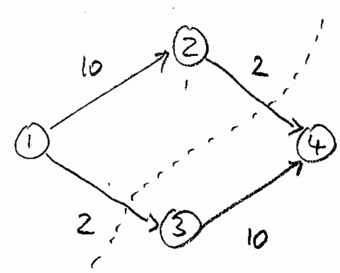
$\Rightarrow t_{max} = 15 \quad t_{min} = 9$

all jobs at maximum duration



$C_{max} = 15$

D_{crit}



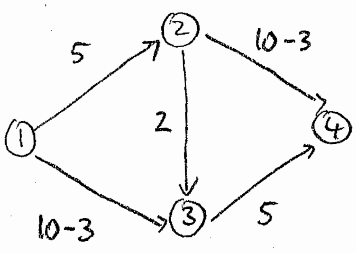
$\delta_1 = 3$ for (2,3)

cost rate 4

$\delta_2 = 4$

$\delta_3 = \infty$

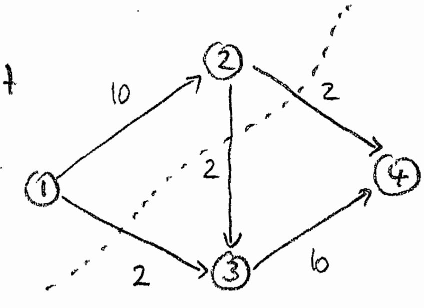
$\Rightarrow \delta = 3$



$C_{max} = 12$

$\delta_1 = \infty$

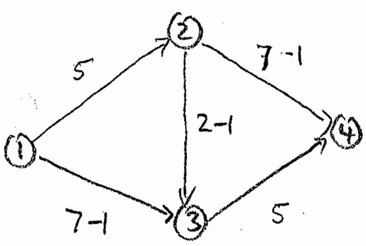
Dcrit



cost rate 6

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

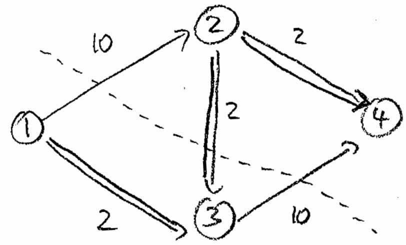
$\Downarrow \; \delta = 1$



$C_{max} = 11$

$\delta_1 = \infty$

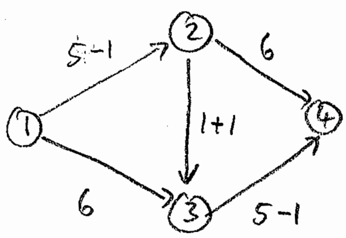
" $\Rightarrow$ " cannot be shortened any more



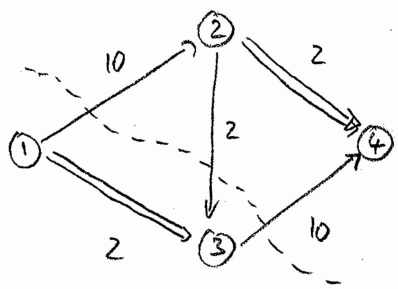
cost rate $10 + 10 - 2 = 18$

$\delta_2 = 2 \qquad \delta_1 = 1$

$\Downarrow \; \delta = 1$



$C_{max} = 10$



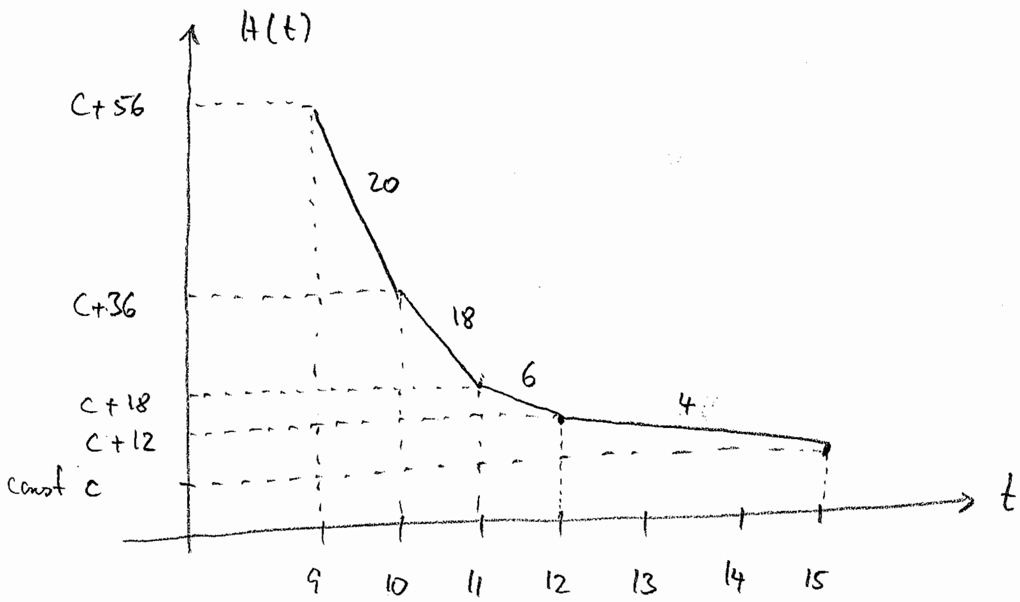
cost rate $10 + 10 = 20$

$\delta_2 = 1 \qquad \delta_3 = \infty$

$\Downarrow \; \delta = 1$

$t_{min} = 9$ reached

NOTE: After shortening, job (2,3) is no longer critical



Question: How to find the good cuts algorithmically

Idea: Use network flows

<u>Max flow - min cut Theorem</u>

> The maximum value of a feasible flow
> = the minimum capacity of a cut

feasible 1, m - flow: flow conservation at every node $i \neq 1, m$
capacities respected $l_{ij} \leq x_{ij} \leq u_{ij}$

lower capacity upper capacity
of arc (i,j)

value of a flow = net outflow in source 1

$$= \sum_j x_{1j} - \sum_j x_{j1}$$

Capacity of cut $[S,T]$

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

$$\operatorname{costrate}(S,T) = \sum_{\substack{(i,j) \text{ forward arc} \\ x_{ij} > a_{ij}}} c_{ij} - \sum_{\substack{(r,s) \text{ backward arc} \\ x_{rs} < b_{rs}}} c_{rs}$$

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]

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

[(i,j) can be lengthened]

[will not contribute to cost rate]

Then (from flow theory)

- Any maximum flow algorithm producing good cuts will find a maximum flow and a (good) cut of minimum capacity
- Here: find flow augmenting paths by BFS
  BFS guarantees (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

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

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 $O(\underbrace{\#\text{cuts calculated}}_{\geq \#\text{breakpoints}} \cdot n^2 \log n)$

is it exponentiell? (YES: EXERCISE)

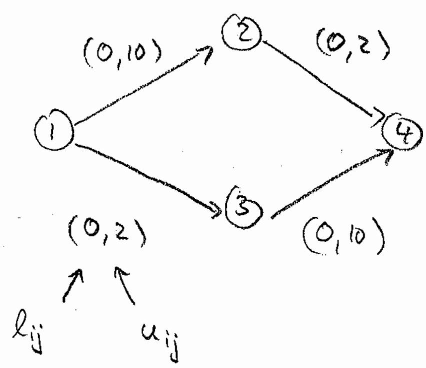
Proof: (1): flow theory

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

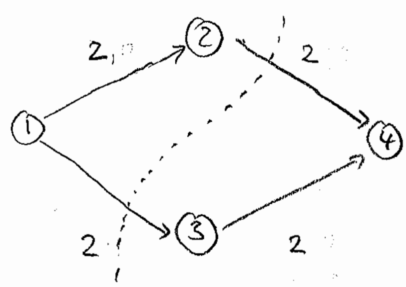
(3) obvious □

19.3 EXAMPLE (continued)

1st flow network

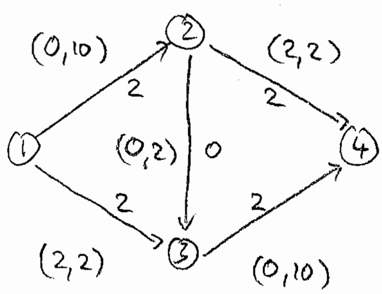


max flow and min cut

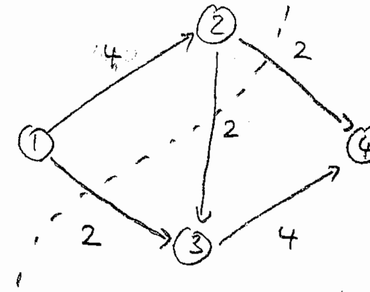


4

2^{nd} flow network

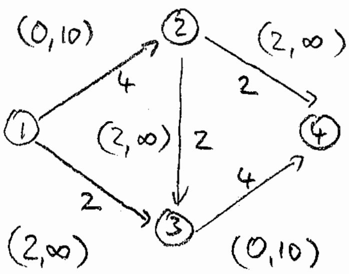


= 0

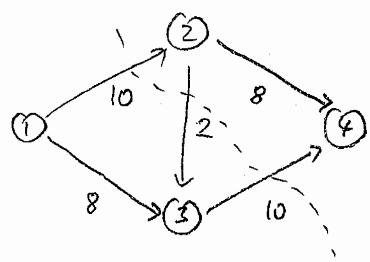


6

3^{rd} flow network

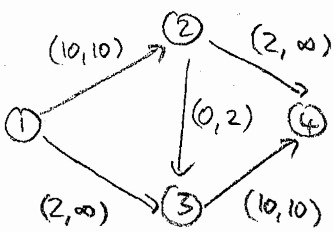


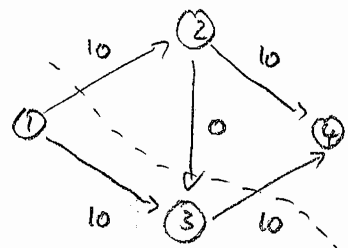
= 0



18

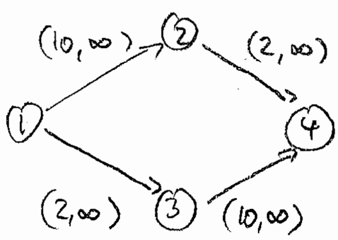
4^{th} flow network

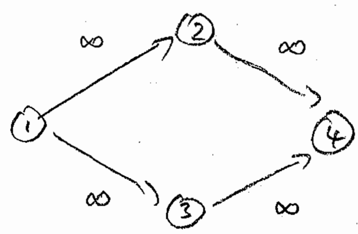




20

5^{th} flow network





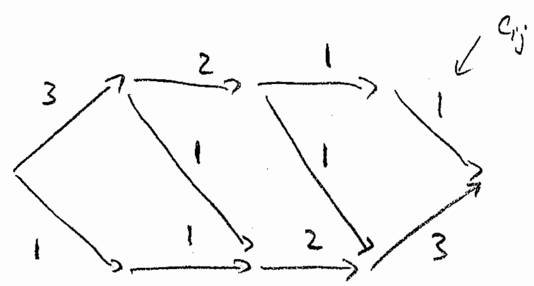
∞

$\Downarrow$

t_{min} reached

Proof of Theorem 19.1 [Sketch on an example]

Expl.



every job has

$a_{ij} = 1 \quad b_{ij} = 10$

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

Problem: an optimal processing time vector for $t - \rho$ may be obtained by shortening several jobs scattered over the whole network

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

Expl: $x = (10, \ldots, 10)$ is optimal for $t = 40$ $(= t_{max})$

Consider $\bar{\delta}$ defined as δ, but over all arcs of D_{crit} (not on a cut)

Expl: $\bar{\delta}_1 = \infty$ (all arcs are critical)

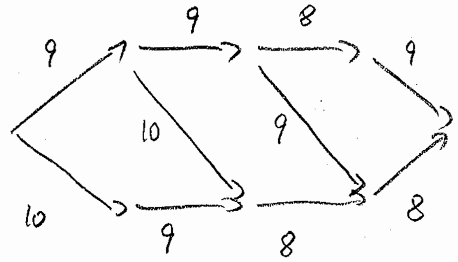
$\bar{\delta}_2 = 9$ (all arcs can be decreased by 9) $\Rightarrow \bar{\delta} = 9$

$\bar{\delta}_3 = \infty$ (no arc can be increased)

Let $\rho_0 \in]0, \bar{\delta}]$ and let y be an opt. proc. time vector for $t - \rho_0$

Expl: $\rho_0 = 5$

y:

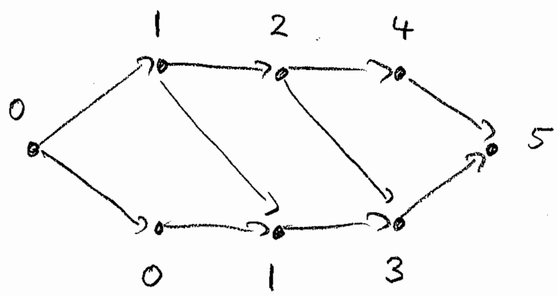


$t - g = 35$

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

Let $\Delta\pi_1 < \Delta\pi_2 < \ldots < \Delta\pi_\ell$ be the different $\delta\pi_i$ values of all i.

Expl:

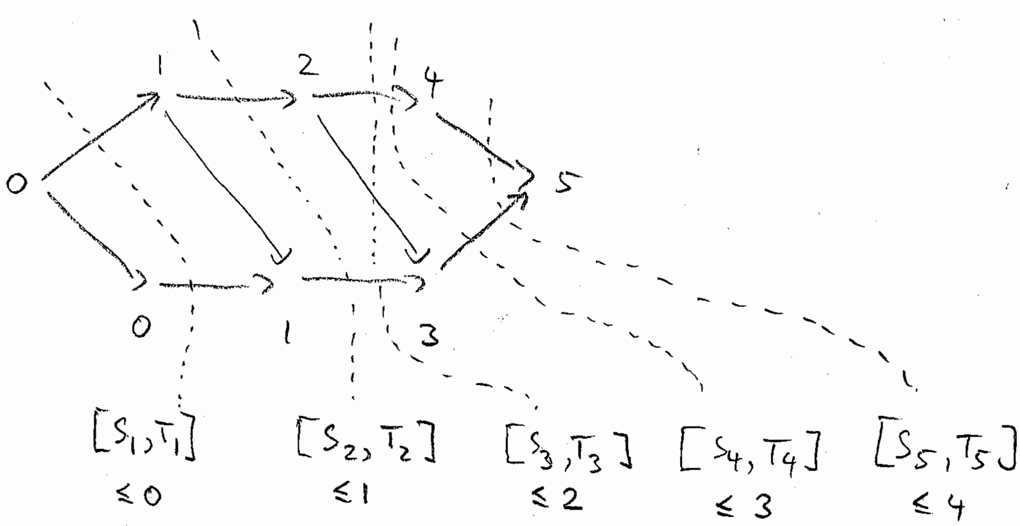


$\leftarrow \delta\pi_i$

$$\begin{array}{cccccc} \Delta\pi_1 & < \Delta\pi_2 & < \Delta\pi_3 & < \Delta\pi_4 & < \Delta\pi_5 & < \Delta\pi_\ell \\ 0 & 1 & 2 & 3 & 4 & 5 \end{array}$$

(1) $[S_k, T_k]$ with $S_k = \{ i \in V \mid \delta\pi_i \le \Delta\pi_k \}$, $T_k = V \setminus S_k$
is a good cut for $k = 1, \ldots, \ell - 1$

Expl:



(2) The following transformation transforms x into y [in D_{crit}]

For $k := 1$ to $\ell-1$ do

(i) $\Delta_k := \Delta\pi_{k-1} - \Delta\pi_k$

(ii) $x_{ij} := x_{ij} - \Delta_k$ for all forward arcs (i,j) in $[S_k, T_k]$

(iii) $x_{rs} := x_{rs} + \Delta_k$ for all backward arcs (r,s) in $[S_k, T_k]$

Expl: $\Delta_k = 1$ for all k, no backward arcs

$\Rightarrow$ subtract 1 on forward arcs of every cut

$\Rightarrow y$

(3) Let $[S_r, T_r]$ be the cut with smallest cost rate among the $[S_i, T_i]$

Let z be the proc. time vector obtained by changing x on $[S_r, T_r]$ by ϱ_0 (possible since $\varrho_0 \le \bar{\delta}$)

Then z is optimal for $t - \varrho_0$

Proof: change of total cost:

For $x \to y$: $\displaystyle\sum_{k=1}^{\ell-1} \text{costrate}[S_k, T_k] \cdot \Delta_k$ because of (2)

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

$$= \text{costrate}[S_r, T_r] \cdot \varrho_0 \quad \hat{=} \quad x \to z$$

(4) Decrease on cut $[S_1, T_1]$ is optimal for <u>all</u> $\varrho_0 \in]0, \bar{\delta}]$

[ϱ_0 has been fixed so far]

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

according to (3). Assume w.l.o.g. that $\varrho_1 \le \varrho_2$

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

[$\Rightarrow$ may use same cut for ϱ_1 and ϱ_2]

If "<", then the decrease on $[\bar{S}_1, \bar{T}_1]$ by $\min\{\varrho_1, \varrho_2\} = \varrho_1$

gives a better solution for $t - \varrho_1$, a contradiction.

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

Now by at most δ (defined by the cut $[S, T]$ of Thm. 19.1 with minimum cost rate)

$\bar{\delta} \le \delta \overset{(4)}{\Rightarrow} [S, T]$ is optimal for a decrease by $\bar{\delta}$ and gives $\bar{z}$

All cuts that can be decreased at time $t - \bar{\delta}$ and $\bar{z}$

can also be decreased at time t and x'.

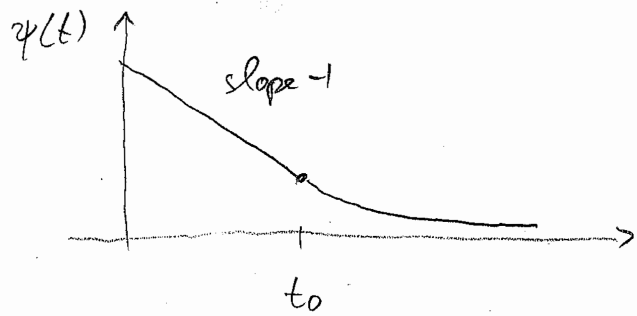
$\Rightarrow [S, T]$ is still the best at $t - \bar{\delta}$ and now

can be further decreased until δ $\square$

Back to the computation of $\psi(t) = \min \sum_{i,j} E[(X_{ij} - x_{ij})^+]$

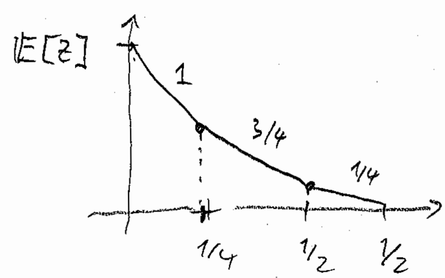
s.t. $C_{max} \le t$

for $t \geq t_0$



Thm 18.2 b) $\Rightarrow$ $\exists$ random variable Z with $\psi(t) = E[(Z-t)^+]$

$\Rightarrow$ first piece of $\psi(t)$ has slope -1
(a property of expected tardiness)



$\hat{=}$ prob for that point

So in the $t \ge t_0$ computation with cost functions $K_{ij}(x_{ij}) = E[(X_{ij} - x_{ij})^+]$
stop when the cost rate of the computed min cut is > 1.
and set the slope to the left of the current time t_0 to -1

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.