# Basic protection/restoration mechanisms and notation

(Pioro - Médhi : 9.1)

- A <u>failure state (scenario)</u> $s$ is a vector of <u>link availability coefficients</u>

$$\alpha^s = (\alpha^s_{(i_1,j_1)}, \alpha^s_{(i_2,j_2)}, \dots \alpha^s_{(i_m,j_m)}) \quad m = |A|$$

where $0 \leq \alpha^s_{(i,j)} \leq 1$ is the proportion of the normal capacity $u_{ij}$ of link $(i,j)$ that is available in scenario $s$, with $s = 1, \dots S$ <u>predefined list of failure scenarios</u>

- Failure scenario's are also characterized by

$$\chi^s = (\chi^s_{d_1}, \dots \chi^s_{d_{|D|}})$$

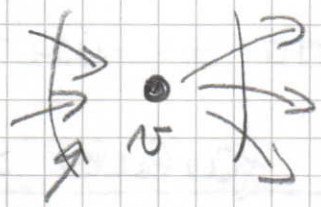
where $\chi^s_d$ is the proportion of $h_d$ (= traffic volume of commodity $d$) that must be realized in scenario $s$ (it may be $\chi^s_d < 1$ meaning decreased realized traffic volume for $d$ under $s$)

- $s = 0$ will be used to denote the <u>normal state</u> (no failures):

  - $\alpha^{0}_{(i,j)} = 1 \quad \forall (i,j) \in A$
  - $\chi^{0}_{d} = 1 \quad \forall d \in D$

<u>example</u>

- <u>failure of a node $v$</u>:




$$\alpha^{s}_{(i,v)} = 0 \quad \forall (i,v) \in BS(i)$$

$$\alpha^{s}_{(v,i)} = 0 \quad \forall (v,i) \in FS(i)$$

$$\chi^{s}_{d} = 0 \quad \forall d = (v,t) \text{ or } d = (t,v)$$

- <u>dynamic (multi-hour) networks</u> (no failures):

$$\forall d \in D \quad h^{s}_{d} = \chi^{s}_{d} \cdot h_{d} \qquad s = \overbrace{1,..S}^{\text{time}}$$

gives the traffic volume of $d$ for

different times of the day (i.e. s now denotes time slot) : in this context it may be $X_d^s > 1$ for some s.

- Protection versus restoration ≡ concerns how resources are re-established in case of failure

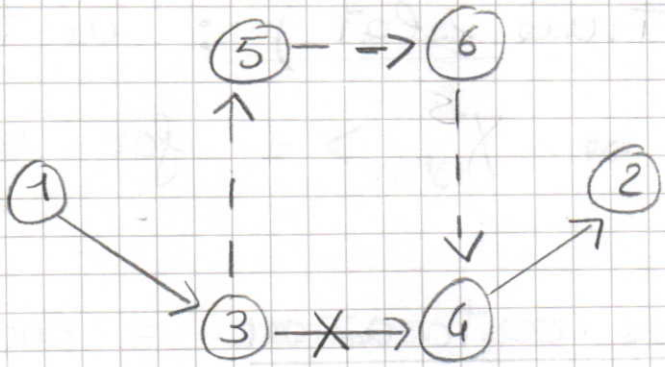
  - protection : mechanisms to "protect" the network "before" the failure happens
    e.g. path diversity

< these are robust approaches >

  - restoration : mechanisms indicating actions to be performed "after" the failure (to try to re-establish resources)

* The term re-establishment is generally used for both protection and restoration *

- Link versus path re-establishment

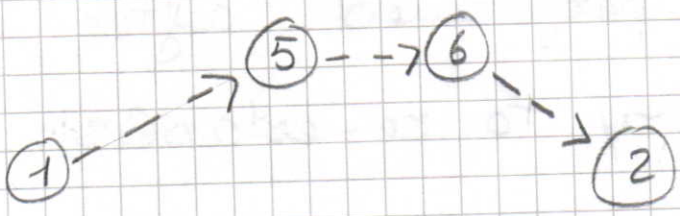



link re-establishment (LR)

assume (1, 3, 4, 2) is used for commodity (1, 2), and link (3, 4) fails :

- link re-establishment : looks for a path from 3 to 4 (e.g. (3, 5, 6, 4)) to re-establish the flow along (3, 4)

- path re-establishment : a new path from 1 to 2 (e.g. (1, 5, 6, 2)) can be used to send flow from 1 to 2




path re-establishment (PR)

- Dedicated versus shared capacity

  - dedicated : spare capacity that is reserved exclusively for re-establishing a certain link or a certain path (and not used in the normal network state)
    - typically used in protection
    - simple to operate
    - costly

  - shared : a common pool of resources is used for re-establishing links and/or paths (also normal capacities)
    - typically used in restoration
    - more complicated to control
    - less expensive

- Splittable versus unsplittable

re-establishment : link or path re-established using several back-up paths or exactly one back-up path.

Protection by Diversity

The simplest way to get protection is to split traffic demands into several link or node disjoint paths : in case of failure a certain portion of traffic demands will survive:

$P_d$ : link or node disjoint candidate paths for $d$, $\forall d \in D$

$n_d$ : diversity factor for $d$, $\forall d \in D$

( $\geq n_d$ different paths with flow $\leq h_d / n_d$ $\forall d \in D$ )

$$\sum_{p \in P_d} x_{dp} = h_d \qquad \forall d \in D$$

$$\sum_{d \in D} \sum_{p \in P_d} \delta_{ij}(dp)\, x_{dp} \leq u_{ij} \qquad \forall (i,j) \in A$$

$$x_{dp} \leq h_d / n_d \qquad \forall d \in D,\ p \in P_d$$

- already studied (4.2.1).

Obs: in case of a single link (node) failure, at most $\frac{100}{n_d}$ % of the traffic demand of d is lost, while the rest survives.

## Link Restoration (LR)

- Assumption: single, total failure of individual links
- Restore entire (or part of) capacity of the failed link (i,j) on one or several paths from i to j
- Spare capacity: shared

## Notation

$$\beta_{vw}((i,j)q) = \begin{cases} 1 & \text{if } (v,w) \text{ belongs to path } q \\ & \text{restoring link } (i,j) \\ 0 & \text{otherwise} \end{cases}$$

$\forall (v,w) \neq (i,j) \in A$

$q \in Q_{ij}$ → candidate restoration paths for $(i,j)$

## Variables

$x^0_{dp}$ : flow in the normal state } normal state

$y_{ij}$ : normal capacity of $(i,j)$ }

$z_{(i,j)q}$ : flow restoring normal capacity of $(i,j)$ on restoration path $q$, $\forall (i,j) \in A$, $q \in Q_{ij}$ } failure

$y'_{ij}$ : spare (≡ protection) capacity of $(i,j)$ (not used in the normal state), $\forall (i,j) \in A$ } failure

$$\text{Min} \sum_{(i,j)\in A} c_{ij}\,(y_{ij} + y'_{ij})$$

$$\sum_{p\in P_d} x^{\circ}_{dp} = h_d \qquad \forall d \in D$$

$$\sum_{d\in D}\sum_{p\in P_d} \delta_{ij}(dp)\, x^{\circ}_{dp} \le y_{ij} \qquad \forall (i,j)\in A$$

$$\sum_{q\in Q_{ij}} z_{(i,j)q} = y_{ij} \qquad \forall (i,j)\in A$$

$$\sum_{q\in Q_{ij}} B_{v,w}((i,j)q)\, z_{(i,j)q} \le y'_{vw} \qquad \forall (i,j),(v,w)$$

($y'_{vw}$: shared)

$$(i,j) \neq (v,w)$$

($(v,w)$: single link failure)

Often: the failed link capacity is restored on a single path $q$:

- additional binary variables $u_{(i,j)q}$
- $\sum_{q\in Q_{ij}} u_{(i,j)q} = 1 \qquad \forall (i,j)\in A$
- $z_{(i,j)q} \le M \cdot u_{(i,j)q} \qquad \forall (i,j)\in A,\ q\in Q_{ij}$

($M$: large constant)

## Path protection and restoration (PR)

(in Médhi - Pioro : 9.3)

- Contrary to link re-establishment, PR restores individual flows rather than link capacities, and is not restricted to single-link failures
- not only shared spare capacity, but it also reuses capacity released by failed paths

### Unrestricted Reconfiguration

We use $d^s_{(i,j)}$, $X^s_d$ for $(i,j) \in A$ and $d \in D$, with s scenario (0 is the normal or nominal state), as previously introduced.

# ILP model

$$z = \text{Min} \sum_{(i,j)\in A} c_{ij}\, y_{ij}$$

$$\sum_{p\in P_d} x_{dp}^{s} = \gamma_d^{s} \cdot h_d \qquad \forall d\in D,\ s=0,..\,S$$

$$\pi_{ij}^{s} \qquad \sum_{d\in D}\sum_{p\in P_d} \delta_{ij}(dp)\, x_{dp}^{s} \le \alpha_{(i,j)}^{s}\, y_{ij} \qquad \forall (i,j)\in A,\ s=0,..\,S$$

$$y_{ij} \ge 0 \qquad \forall (i,j)\in A$$

$$x_{dp}^{s} \ge 0 \qquad \forall d\in D,\ p\in P_d,\ s=0,..\,S$$

where $\{x_{dp}^{s}\}$ denote flows in scenario $s$

- Unrestricted reconfiguration since unrestricted flow patterns can be established in each scenario (included $s=0$)
- Flows can therefore be splittable in each scenario

How can we deal with the LP relaxation? Due to the large number of commodities and scenarios, then LP solvers may fail

1) Lagrangian relaxation

If we relax the capacity constraints via Lagrangian multipliers $\overline{\Pi}^{s}_{ij}$, then the resulting problem decomposes into separate shortest path problems, one for each commodity d and each scenario s:

$$z(\overline{\Pi}) = \text{Min} \sum_{(i,j)\in A} \left(c_{ij} - \sum_{s} \overline{\Pi}^{s}_{ij} \alpha^{s}_{(i,j)}\right) y_{ij} +$$

$$+ \sum_{(i,j)\in A} \sum_{d\in D} \sum_{p\in P_d} \sum_{s} \overline{\Pi}^{s}_{ij}\, \delta_{ij}(dp)\, x^{s}_{dp}$$

$$\sum_{p\in P_d} x^{s}_{dp} = X^{s}_{d} \cdot h_d \qquad \forall d\in D,\ s=0..S$$

$$y_{ij} \geq 0 \qquad \forall (i,j)\in A$$

$$x^{s}_{dp} \geq 0 \qquad d\in D,\ p\in P_d,\ s=0..S$$

Note that, given $\{\pi_{ij}^s\}$, then

$$y_{ij} = \begin{cases} 0 & \text{if } c_{ij} - \sum_s \pi_{ij}^s \alpha_{(i,j)}^s \geq 0 \\ +\infty & \text{otherwise} \end{cases}$$

Since $z(\pi) = -\infty$ if $c_{ij} - \sum_s \pi_{ij}^s \alpha_{(i,j)}^s < 0$ for some $(i,j) \in A$, the Lagrangian Dual consists in

$$\text{Max } z(\pi)$$

$$\sum_s \pi_{ij}^s \alpha_{(i,j)}^s \leq c_{ij} \quad \forall (i,j) \in A$$

$$\pi_{ij}^s \geq 0 \quad \forall (i,j) \in A, \; s = 0 \dots S$$

2) Column generation

Also in this case compute a shortest path in $P_d$ for each $d \in D$ and each scenario, w.r.t costs depending on the scenario, by generalizing the approach described for multicommodity flows

< no detail >

# Restricted Reconfiguration

natural restriction: "unbroken" flows are not moved

Now

$\alpha^s_{(i,j)} \in \{0,1\} \quad (i,j) \in A, \; s = 0 \ldots S$

$\theta^s_{dp} = \prod_{(i,j)\in p} \alpha^s_{ij}$ availability of p in scenario s

$$\text{Min} \sum_{(i,j)\in A} c_{ij} y_{ij}$$

< $h^s_d$ is the "minimum" realized volume >

$$\sum_{p \in P_d} x^s_{dp} \geq h^s_d \qquad d \in D, \; s = 0 \ldots S$$

due to

$$\sum_{d\in D} \sum_{p \in P_d} \delta_{ij}(dp)\, x^s_{dp} \leq \alpha^s_{(i,j)}\, y_{ij} \qquad (i,j) \in A, \; s = 0 \ldots S$$

$$x^s_{dp} \geq \theta^s_{dp}\, x^0_{dp} \qquad d \in D, \; p \in P_d \quad s = 1 \ldots S$$

$$x^0_{dp} \geq 0 \qquad d \in D, \; p \in P_d$$

"unbroken" flows are not moved (but it is possible to use unaffected paths for restoring flows from failed paths ($\geq$))

## Protection against traffic demand uncertainty

- often traffic demands are uncertain
- how can we protect the network against such an uncertainty ?

Assumption : $\bar{h}_d - \delta_d \leq h_d \leq \bar{h}_d + \delta_d$ , $\forall d \in D$, where $\bar{h}_d$ may denote the estimated average traffic volume of $d$ ( let denote it by $H_D$)

Objective : to install capacities (at a minimum cost ) so as to support all the traffic demands in $H_D$ , and therefore also the worst case scenario.

Further assumption :

Let $\tilde{x}_{dp}$ be the fraction of the traffic demand of $d$ sent along $p$, $\forall d \in D$, $p \in P_d$

i.e. $\begin{cases} \sum_{p \in P_d} \tilde{x}_{dp} = 1 \\ 0 \leq \tilde{x}_{dp} \leq 1 \quad \forall p \in P_d \end{cases}$ $\forall d \in D$

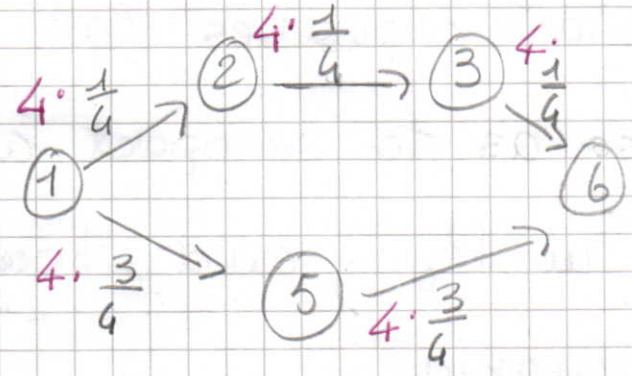
routing template

we impose that, whatever the traffic requirement of $d$, $h_d$, in $H_D$, then the traffic is always sent according to the routing template $\{\tilde{x}_{dp}\}$, i.e.:

$$x_{dp} = h_d \cdot \tilde{x}_{dp} \quad \forall p \in P_d$$

stable or oblivious routing

example




- $\tilde{x}_{dp}$ (routing template)

$d$ is the pair $(1, 6)$

$\frac{1}{4}$ along $P_1 = (1\ 2\ 3\ 6)$ and $\frac{3}{4}$ along $P_2 = (1\ 5\ 6)$

- If $h_d = 4$, then multiply $\tilde{x}_{dp}$ by 4 (in red)

## ILP formulation (?)

$$\min \sum_{(i,\delta) \in A} c_{i\delta}\, y_{i\delta}$$

$$\sum_{p \in P_d} \tilde{x}_{dp} = 1 \qquad \forall d \in D$$

< routing Template >

$$0 \le \tilde{x}_{dp} \le 1 \qquad \forall d \in D,\ p \in P_d$$

$$\max_{h_d \in H_D} \sum_{d \in D} \sum_{p \in P_d} \delta_{i\delta}(dp)\, \underbrace{h_d \cdot \tilde{x}_{dp}}_{x_{dp}} \le y_{i\delta} \qquad \forall (i,\delta) \in A$$

< install capacities for the worst case >

- This is not linear!
- Consider the inner problem $\forall (i,\delta) \in A$ :

$$\max \sum_{d \in D} \sum_{p \in P_d} \delta_{i\delta}(dp)\, h_d\, \tilde{x}_{dp}$$

$$h_d \le \bar{h}_d + \delta_d \qquad \forall d \in D \qquad (w_d)$$

simplified version $\rightsquigarrow$ $h_d \ge 0 \qquad \forall d \in D$

For given $\{\tilde{x}_{dp}\}$, this is a LP;

consider its dual :

$$\min \sum_{d \in D} w_d \left( \bar{h}_d + \delta_d \right)$$

$$w_d \geq \sum_{p \in P_d} \bar{\delta}_{ij}(dp) \cdot \tilde{x}_{dp} \quad \forall d \in D$$

$$w_d \geq 0 \quad \forall d \in D$$

From strong duality, the optimal values are equal, therefore replace each inner problem with its dual:

$$\text{Min} \sum_{(i,j) \in A} c_{ij} y_{ij}$$

$$\sum_{p \in P_d} \tilde{x}_{dp} = 1 \quad \forall d \in D$$

$$0 \leq \tilde{x}_{dp} \leq 1 \quad \forall d \in D, p \in P_d$$

$$\left[ \begin{array}{l} \cancel{\min} \sum_{d \in D} w_d \left( \bar{h}_d + \delta_d \right) \leq y_{ij} \quad \forall (i,j) \in A \\ w_d \geq \sum_{p \in P_d} \bar{\delta}_{ij}(dp) \tilde{x}_{dp} \quad \forall d \in D \\ w_d \geq 0 \quad \forall d \in D \end{array} \right]$$

We can remove the operator "min" (due to the nature of constraints, "$\leq$"), so obtaining a Linear Programming model: easy!

< Robust optimization approach >

## Multi-period network design

(Médhi - Pioro: 11.2.1)

- Capacity planning in the multi-period case

$T$ : time horizon

$P_d^\tau$ : candidate paths for $d$ in period $\tau$,

$d \in D, \tau = 1..T$

$h_d^\tau$ ($\geq 0$): new (incremental) demand volume of $d$ in $\tau$, $d \in D$, $\tau = 1..T$

$c_{ij}^\tau$ : installation cost of one module of capacity on $(i,j)$ at $\tau$

$c'^{\tau}_{ij}$ : maintenance cost of one module of capacity on $(i,j)$ at $\tau$

$M$ : size of one capacity module



## Variables

$x_{dp}^{\tau}$ : flow on $p$ for $d$ at time $\tau$

$y_{ij}^{\tau}$ : number of modules of capacity on $(i,j)$ needed at $\tau$ (installed at $\tau$)

## ILP model

$$\min \sum_{\tau=1}^{T} \sum_{(i,j)\in A} \left( c_{ij}^{\tau} + \sum_{t \geq \tau} c_{ij}^{t} \right) y_{ij}^{\tau}$$

$$\sum_{p \in P_d^{\tau}} x_{dp}^{\tau} = h_d^{\tau} \quad \forall d \in D,\ \tau = 1, \ldots T$$

$$\sum_{d \in D} \sum_{p \in P_d^{\tau}} \delta_{(i,j)}^{\tau}(dp)\, x_{dp}^{\tau} \leq M\, y_{ij}^{\tau} \quad \forall (i,j) \in A,\ \tau = 1 \ldots T$$

$$x_{dp}^{\tau} \geq 0 \quad \forall d \in D,\ p \in P_d^{\tau},\ \tau = 1, \ldots T$$

$$y_{ij}^{\tau} \geq 0, \text{ integer} \quad \forall (i,j) \in A,\ \tau = 1, \ldots T$$

- This model can be decomposed into $T$ independent ILP models (one per period of time!)

## Main drawbacks

1) $h_d^\tau < 0$ could be possible! (reduced traffic volume)

2) not all capacity installed at periods preceeding $\tau$ is used at $\tau$: some spare capacity could be available to one period to the next which could be used! (this issue is not addressed by the previous model).

By incorporating observation 2), we can define:

## Capacity Reuse model

assumption: $h_d^\tau \geq 0$, $d \in D$, $\tau = 1, \dots T$

in addition: $\xi_{dp}^\tau$ unit routing cost on $p \in P_d^\tau$, $d \in D$, $\tau = 1, \dots T$

$$z = \min \sum_{t=1}^{T} \sum_{(i,j)\in A} \left( c_{ij}^{t} + \sum_{\tau \geq t} c'^{\,t}_{ij} \right) y_{ij}^{t} +$$

$$+ \sum_{t=1}^{T} \sum_{d\in D} \sum_{p\in P_d^t} q_{dp}^{t} \, x_{dp}^{t}$$

$$\sum_{p\in P_d^t} x_{dp}^{t} = h_d^{t} \qquad d\in D,\ t = 1,\dots T$$

$$\sum_{\tau \leq t} \sum_{d\in D} \sum_{p\in P_d^\tau} \delta_{ij}^{\tau}(dp)\, x_{dp}^{\tau} \leq \sum_{\tau \leq t} M\, y_{ij}^{\tau} \qquad \forall (i,j)\in A,\ t = 1,\dots T$$

$$x_{dp}^{t} \geq 0 \qquad d\in D,\ p\in P_d^t,\ t = 1,\dots T$$

$$y_{ij}^{t} \geq 0 \text{ integer} \qquad (i,j)\in A,\ t = 1,\dots T$$

- total (cumulative) flow sent on $(i,j)$ at $t$

If we want to "rearrange" the routing from one period to the next one:

$$\sum_{p\in P_d^t} x_{dp}^{t} = \sum_{\tau \leq t} h_d^{\tau} \qquad d\in D,\ t = 1,\dots T$$

$$\sum_{d\in D} \sum_{p\in P_d^t} \delta_{ij}^{t}(pd) \cdot x_{dp}^{t} \leq \sum_{\tau \leq t} M\, y_{ij}^{\tau} \qquad (i,j)\in A,\ t = 1,\dots T$$

# A dynamic programming approach for solving the Capacity Reuse model

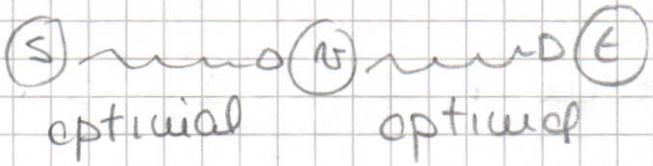
## On Dynamic Programming

(Wolsey : 5.1 , 5.4.1)

Some motivations : shortest paths

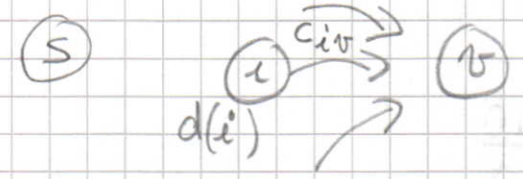
- $G = (N, A)$
- $s$ origin
- $c_{ij} \geq 0 \quad \forall (i,j) \in A$

Observation 1 : if a shortest path from $s$ to $t$ passes by node $v$, then the subpaths from $s$ to $v$ and from $v$ to $t$ are shortest paths from $s$ to $v$ and from $v$ to $t$, respectively




Observation 2 : if $d(v)$ denotes the shortest path length from $s$ to $v$, then :

$$d(v) = \min_{(i,v) \in BS(v)} \{ d(i) + c_{iv} \} \qquad (*)$$




For acyclic graphs, (*) provides a simple algorithm ($O(m)$): if $s = 1$, and the nodes are ordered so that $(i,j) \in A \rightarrow i < j$, then apply the recurrence (*) for $v = 2, \dots n$.

For general graphs? to impose an ordering, define $D_k(i)$ as the length of the shortest path from s to i among the paths containing at most k arcs. Then:

$$D_k(v) = \min \{ D_{k-1}(v), \min_{(i,v) \in BS(v)} ( D_{k-1}(i) + c_{iv} ) \}$$

By increasing k from 1 to $(n-1)$, and at each time calculating $D_k(v)$ $\forall v \in N$ by the recursion above, at termination $d(v) = D_{n-1}(v)$ $\forall v \in N$ : $O(mn)$

This is an example of Dynamic Programming (DP): the optimal solution value is calculated recursively from the optimal values of slightly different problems

Standard terminology in (DP):

- Principle of Optimality : when "pieces" of optimal solutions are themselves optimal (as in Shortest paths)
- States : correspond to the nodes for which values need to be calculated ($D_k(v)$ for given $k$ in Shortest paths)
- Stages : steps which define the ordering ($k = 1, .. (n-1)$ in Shortest paths)

(DP) apply to other interesting problems; one example is 0-1 Knapsack

# (DP) for 0-1 Knapsack

$$z = \max \sum_{j=1}^{n} c_j x_j$$

(KP)
$$\sum_{j=1}^{n} a_j x_j \leq b$$
$$x_j \in \{0,1\} \quad j = 1, \dots n$$

Define $z_r(\lambda) = \max \sum_{j=1}^{r} c_j x_j$

smaller
(KP)$_s$
$$\sum_{j=1}^{r} a_j x_j \leq \lambda$$
$$x_j \in \{0,1\} \quad j = 1, \dots r,$$

where $\lambda = 0, 1, \dots b$, $r = 0, \dots n$.

Clearly $z = z_n(b)$.

We can then define the following recursion:

$$\boxed{z_r(\lambda) = \max \{ z_{(r-1)}(\lambda), c_r + z_{(r-1)}(\lambda - a_r) \}}$$

if $(\lambda - a_r) \geq 0$

Then:

- start with $z_0(\lambda) = 0 \quad \forall \lambda \geq 0$
- use the recursion to calculate $z_2(\lambda)$, $\lambda = 0, 1, \dots b$, $z_3(\lambda)$, $\lambda = 0, 1, \dots b$, $\dots z_n(\lambda)$, $\lambda = 0, 1 \dots b$

States : $z_k(\lambda)$ for $\lambda = 0, 1, \dots b$ for given $k$

Stages : $k = 1, \dots n$

$O(n \cdot b)$ pseudopolynomial

example

$$z = \max 10x_1 + 7x_2 + 25x_3 + 24x_4$$
$$2x_1 + x_2 + 6x_3 + 5x_4 \leq 7$$
$$x_1, x_2, x_3, x_4 \in \{0, 1\}$$

| | $z_1$ | $z_2$ | $z_3$ | $z_4$ | $P_1$ | $P_2$ | $P_3$ | $P_4$ |
|---|---|---|---|---|---|---|---|---|
| $\lambda=0$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 7 | 7 | 7 | 0 | 1 | 0 | 0 |
| 2 | 10 | 10 | 10 | 10 | 1 | 0 | 0 | 0 |
| 3 | 10 | 17 | 17 | 17 | 1 | 1 | 0 | 0 |
| 4 | 10 | 17 | 17 | 17 | 1 | 1 | 0 | 0 |
| 5 | 10 | 17 | 17 | 24 | 1 | 1 | 0 | 1 |
| 6 | 10 | 17 | 25 | 31 | 1 | 1 | 1 | 1 |
| 7 | 10 | 17 | 32 | 34 | 1 | 1 | 1 | 1 |

For example, $z_2(7) = \max\{z_1(7), \overset{c_2}{\overset{\|}{7}} + z_1(7 - \overset{a_2}{\overset{\|}{1}})\} =$

$= \max\{10, 7 + 10\} = 17.$

- The optimal value, $z$, is

$$z = z_4(7) = 34.$$

- Values $p_r$ ($p_r(\lambda)$) are used to recover an optimal solution:

$$p_r(\lambda) = \begin{cases} 0 & \text{if } z_r(\lambda) = z_{(r-1)}(\lambda) \\ 1 & \text{otherwise} \end{cases}$$

$$r = 1, \dots, n$$
$$\lambda = 0, \dots b$$

example : $p_2(7) = 1$

- They are used back from $p_n(b)$

example :

$p_4(7) = 1 \rightarrow x_4^* = 1$

$p_3(7 - \underbrace{5}_{a_4}) = p_3(2) = 0 \rightarrow x_3^* = 0$

$p_2(2) = 0 \rightarrow x_2^* = 0$

$p_1(2) = 1 \rightarrow x_1^* = 1$

optimal solution

Note that we started the recursion with :

- $z_1(\lambda) = 0$ for $0 \leq \lambda < a_1$
- $z_1(\lambda) = c_1$ for $\lambda \geq a_1$

instead of $z_0(\lambda) = 0$ for $\lambda \geq 0$.

## A (DP) approach for capacity

### Reuse model

(Médhi - Pioro : 11.2.5)

Assumption 1 : $c'^{z}_{ij} = 0$ , $\rho^{z}_{dp} = 0$ $(i,j) \in A$, $d \in D$, $p \in P_d$, $z = 1, .. T$

Assumption 2 : $x^{z}_{dp}$ integer (integer flows)

Assumption 3 : $P^{z}_{d}$ do not depend on $z$

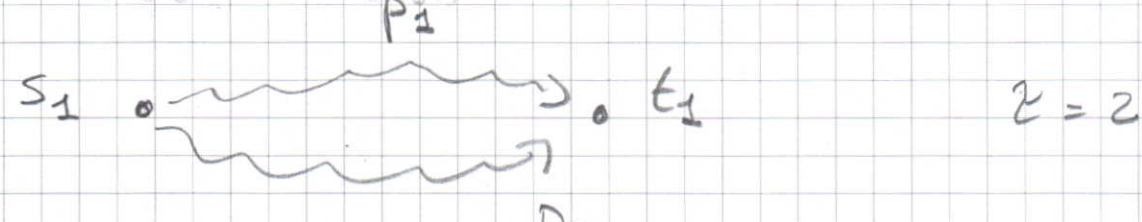
**Stages** : time periods $z = 1, .. T$
(optimize the capacity expansion in the next period)

**Steps** : $f_{\delta}^{z} = (\underbrace{\quad \cdots \quad}_{P_{d_1}} \underbrace{\quad \cdots \quad}_{P_{d_{|D|}}}, f)$,
(states)

where $f_{\delta}^{z}(p)$ gives the units of flow along $p$ to satisfy the traffic volumes from $t = 1$ to $t = z$ ($f$ may be exponential)

**Then**, if $C(f_{\delta}^{z})$ is the minimum capacity cost related to $f_{\delta}^{z}$, we have the following kind of recurrence :

**example**




$$h^{1}_{(s_1 t_1)} = 2 \qquad h^{2}_{(s_1, t_1)} = 1$$

possible state for $z = 2$ :

$$f_j^z = \begin{pmatrix} 2 & 1 \\ P_1 & P_2 \end{pmatrix}$$

$f_j^z$ can be obtained from the states of $z = 1$ :

$$(1 \quad 1) \qquad (2 \quad 0)$$

Therefore

$$c(f_j^z) = c(2, 1) = \min \begin{cases} c(1,1) & \text{if the installed capacity allows to send 1 unit along } P_1 \\ c(1,1) + \text{cost additional capacity on } P_1, \text{ otherwise} \\ c(2,0) & \text{if the installed capacity allows to send 1 unit along } P_2 \\ c(2,0) + \text{cost additional capacity on } P_2, \text{ otherwise} \end{cases}$$