On the notion of max-min fairness and its applications to routing optimization

Michał Pióro

Institute of Telecommunications
Warsaw University of Technology (Poland)

Department of Electrical and Information Technology
Lund University (Sweden)

HET-NETs 2010, Zakopane, Poland

motivation

- **fairness is an important objective in communication network design**
  - **yet not commonly understood**

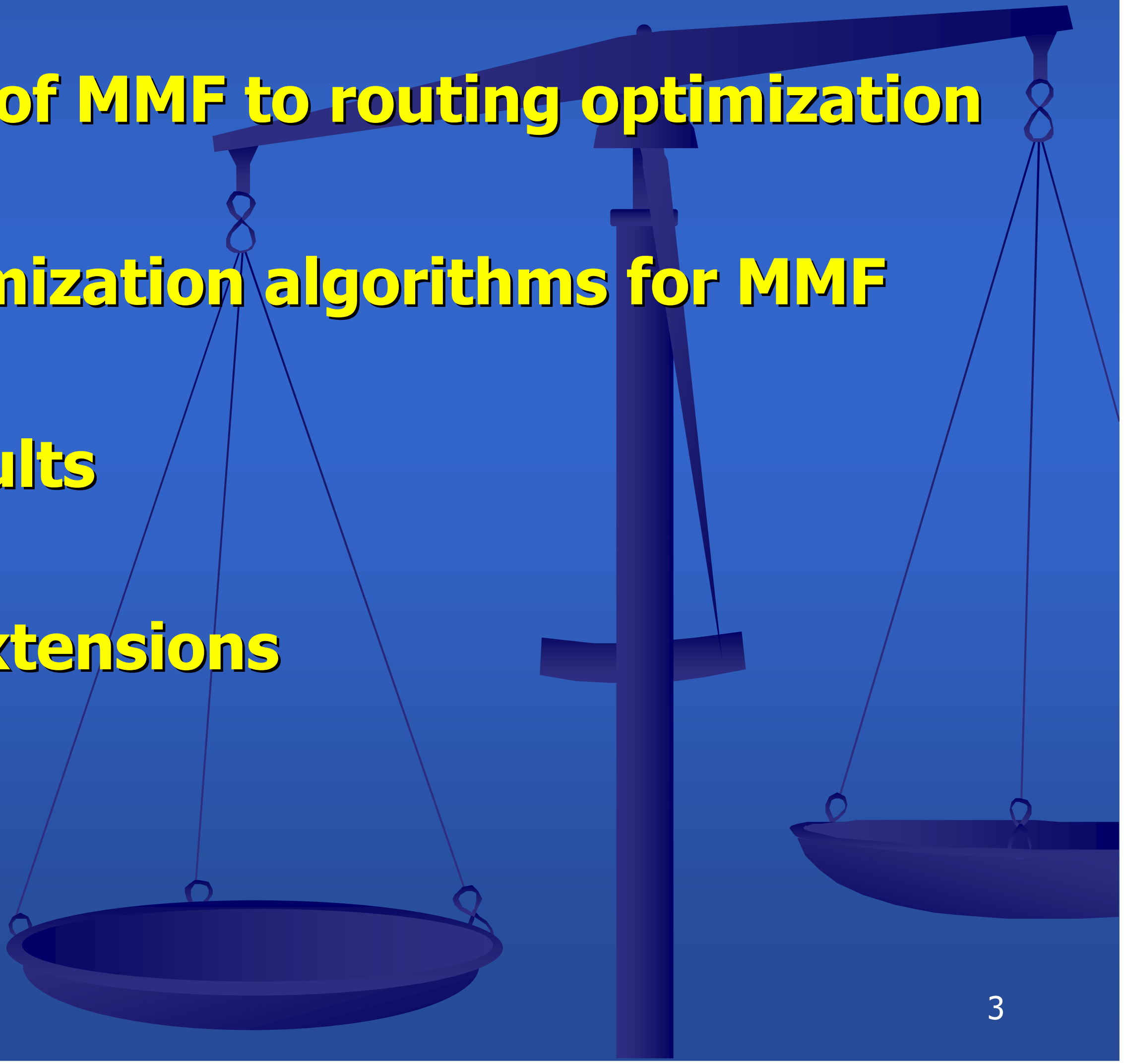
- **example applications**
  - **routing of elastic traffic in the Internet**
  - **resource utilization or resource distribution**
  - **design of resilient networks**

- **there are different notions of fairness**
  - **MMF – max-min fairness**
  - **PF – proportional fairness**

- **optimization methods for problems involving fairness are hardly known to researchers in telecommunications**
  - **MMF is frequently „re-invented" (often in a wrong way)**

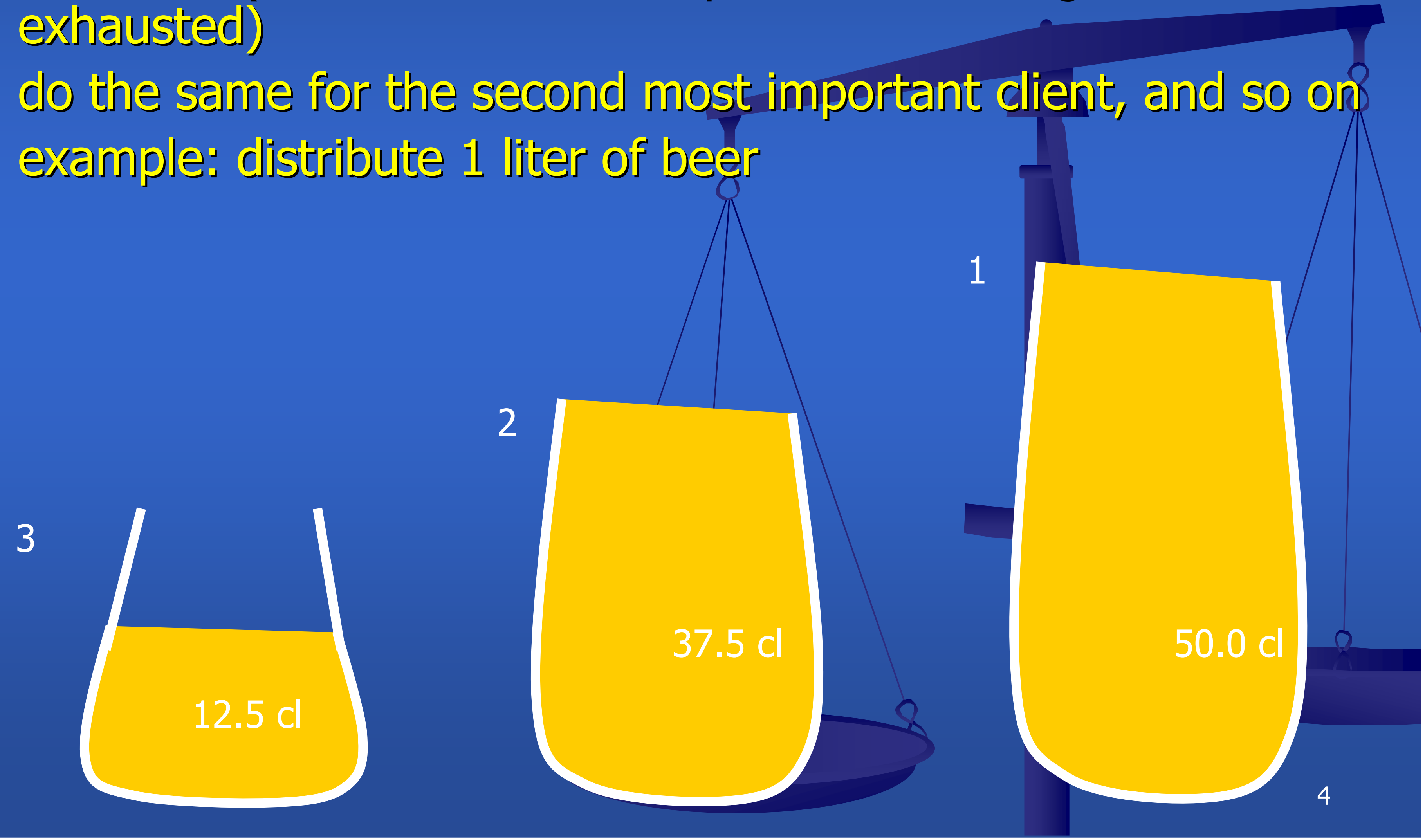
purpose of the presentation (and outline)

- introduce the notion of MMF
- show applications of MMF to routing optimization
- present basic optimization algorithms for MMF
- show example results
- discuss selected extensions



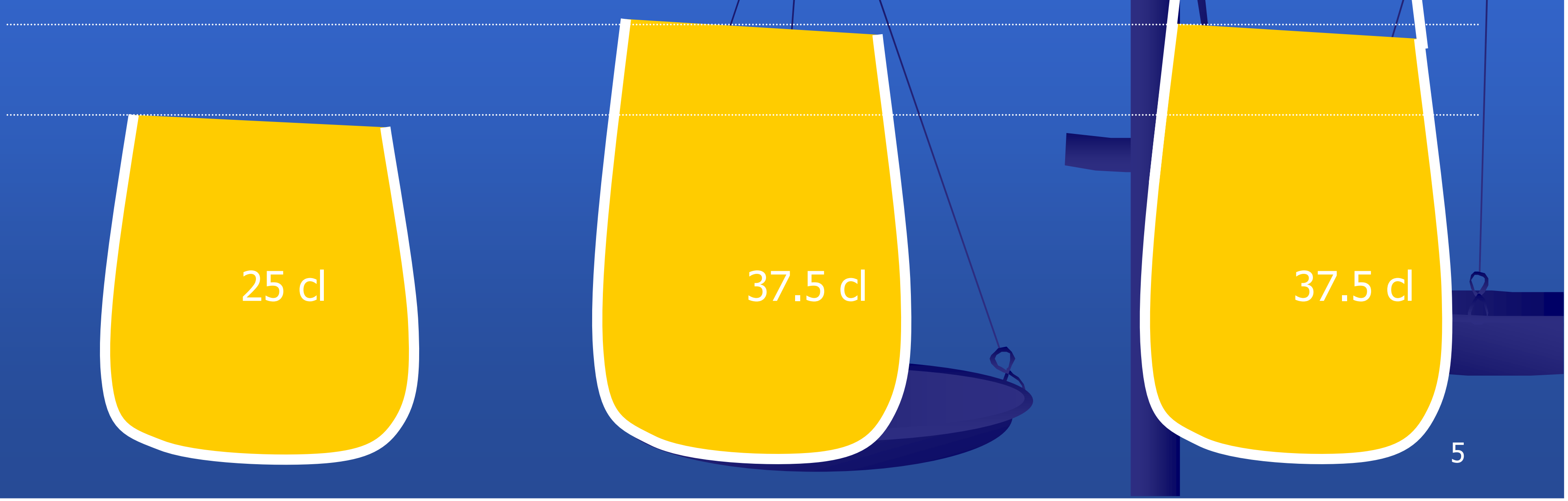
lexicographically maximal solution (priority of customers)

- give as much goods as possible to the most important customer (until he cannot accept more, or the goods are exhausted)
- do the same for the second most important client, and so on
- example: distribute 1 liter of beer



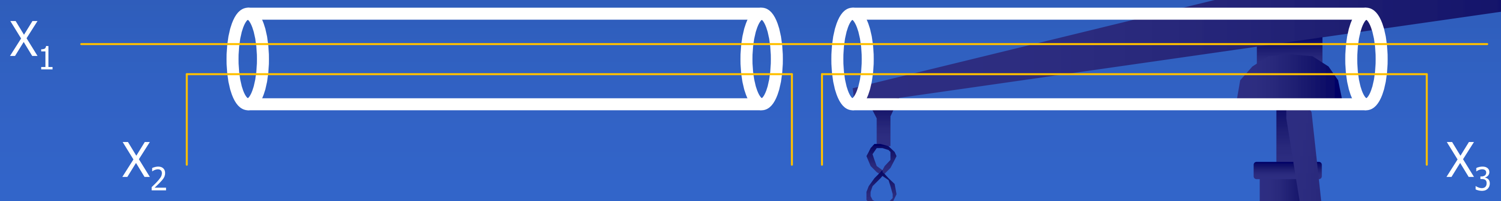
max-min fairness: beer distribution

- give as much goods as possible equally to all, until
  - one customer cannot accept more or
  - the goods are exhausted
- if there are more goods left, distribute them equally to those who are still able to receive them
- and so on, until either no one can accept more, or the goods are exhausted
- example of a MMF solution:
  distribute 1 liter of beer



routing problem for a simple network

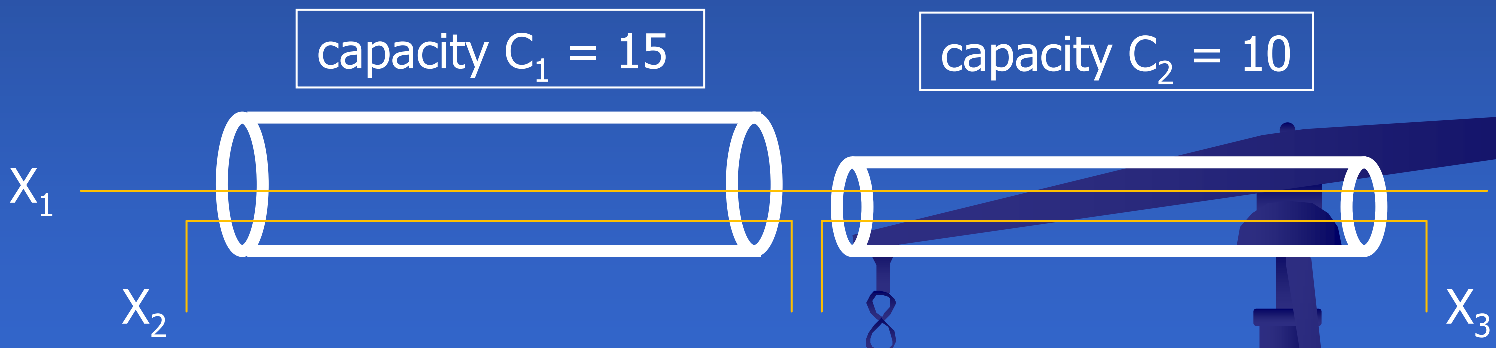
- two links in series – each of capacity 10 (e.g., 10 Mbps)
- three elastic demands (flows) eager to get as much bandwidth as possible



throughput

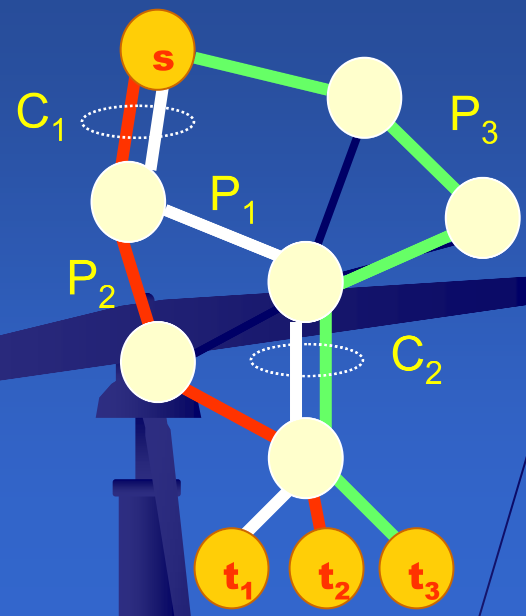
- lexicographical fairness (priority 3,2,1): $X_1 = 0$, $X_2 = X_3 = 10$ (20)
  - maximize X_3, then maximize X_2 with fixed X_3, finally maximize X_1 with fixed X_2 and X_3
- max-min fairness: $X_1 = X_2 = X_3 = 5$ (15)
  - maximize the worst flow (next worst flow, …)

routing problem for a simple network (modified)



- lexicographical fairness (priority 3,2,1): $X_1 = 0$, $X_2 = 15$, $X_3 = 10$ (25)
- max-min fairness: $X_1 = 5$, $X_2 = 10$, $X_3 = 5$ (20)

routing problem



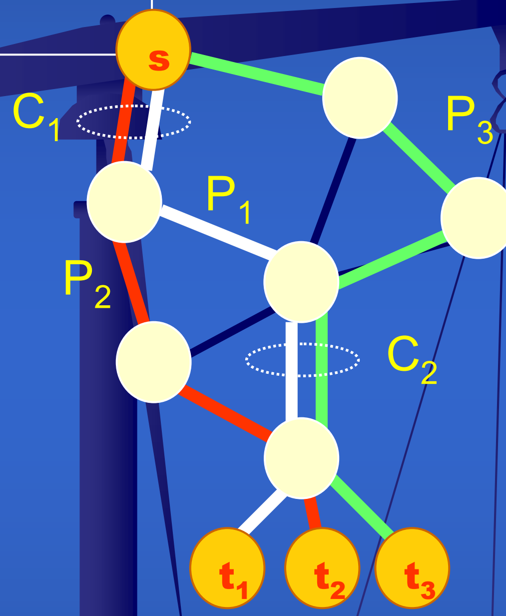
- C_e – capacity of link e, $e \in E$
- three connections (d = 1,2,3) corresponding to three fixed paths from s to t_1, t_2, t_3
- assign bandwidth X_1, X_2, X_3 respectively to paths P_1, P_2, P_3 in a fair way

- $\sum_{\{d:\, e \in Pd\}} X_d \leq C_e$, $e \in E$ (capacity constraint)
  - for example: $X_1 + X_2 \leq C_1$, $X_1 + X_3 \leq C_2$

elastic traffic: the amount of goods accepted by a connection is potentially infinite

lexmax routing – solution

$C_1 = 15$ $C_2 = 10$ (bottleneck links)
the rest of links have large capacity
the amount of goods (bandwdth) is infinite



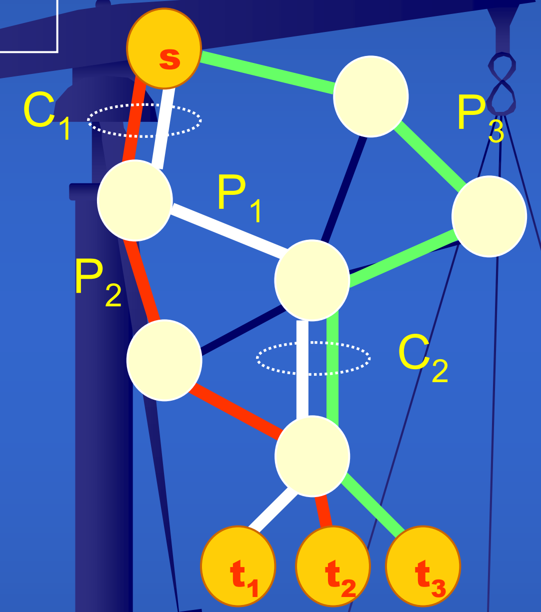
- $X_1 + X_2 \leq 15$
- $X_1 + X_3 \leq 10$
- importance of connections: 3, 2, 1

solution:

- step 1: $X_3 = 10$ (link 2 gets saturated)
- step 2: $X_2 = 15$ (link 1 gets saturated)
- step 3: $X_1 = 0$ (link 1 and link 2 are saturated)
- finally: $X_1 = 0$ $X_2 = 15$ $X_3 = 10$

MMF routing - solution

$C_1 = 10$ $C_2 = 15$ (bottleneck links)
the rest of links have large capacity
the amount of goods (bandwdth) is infinite



- $X_1 + X_2 \leq 15$
- $X_1 + X_3 \leq 10$

solution:

- step 1: $X_1 = X_2 = X_3 = 5$ (link 2 gets saturated)
- step 2: $X_2 = 10$ (link 1 gets saturated)
- finally: $X_1 = 5$ $X_2 = 10$ $X_3 = 5$

algorithm (waterfilling)

(Bertsekas & Gallager „Data Networks”)

$n_e = |\{d \in D: e \in P_d\}|\,,\ \ e \in E$ (number of paths through a link)

Step 0: $X = (X_1, X_2, \ldots, X_D) := 0\ ;\quad k := 0.$

Step 1: $k := k+1$

set $t = \min_{e \in E} C_e / n_e$

for all $e \in E$ put $C_e := C_e - t \cdot n_e$

for all $d \in D$ put $X_d := X_d + t$

remove all saturated links and all connections through the removed links.

Step 2: Stop if there are no connections left; otherwise go to Step 1.

$$\frac{C_e}{n_e}$$

Not so simple in the general case!

basic notations: lexicographical order

- $y = (y_1, y_2, \dots, y_m)$, $z = (z_1, z_2, \dots, z_m)$ vectors in R^m (m-vectors)
- lexicographical order:

$$(y_1, y_2, \dots, y_m) <_{lex} (z_1, z_2, \dots, z_m)$$

iff there exists $0 \le k < m$ such that

- $y_j = z_j$ for $j = 1, 2, \dots, k$
- $y_{k+1} < z_{k+1}$

the rest of entries $(j = k+2, k+3, \dots, m)$ do not matter!

examples: $(1, 2, 1000) <_{lex} (1,3,1)$ $(1, 100, 1000) <_{lex} (2,2,2)$

basic notations: MMF order

- $y = (y_1, y_2, \ldots, y_m)$, $z = (z_1, z_2, \ldots, z_m)$
- MMF order:

$$(y_1, y_2, \ldots, y_m) <_{MMF} (z_1, z_2, \ldots, z_m)$$

iff

$$[\,(y_1, y_2, \ldots, y_m)\,] <_{lex} [\,(z_1, z_2, \ldots, z_m)\,]$$

where $[\,x\,]$ denotes vector x <u>sorted in non-decreasing order</u>

examples: $(1,2,2) <_{lex} (1,3,1)$ $(1,2,3) <_{lex} (1,3,3)$

$(1,3,1) <_{MMF} (1,2,2)$ (because $(1,1,3) <_{lex} (1,2,2)$)

$(1,2,3) <_{MMF} (1,3,3)$ (already sorted)

three problems

- $X \subseteq R^n$ solution space
- $x = (x_1,x_2,\ldots,x_n)$ n-vector, $x \in X$ (variables)
- $f(x) = (f_1(x),f_2(x),\ldots,f_m(x))$, $f_j : X \to R$ (criteria)

Find $x^0 \in X$ such that:

LEXMAX: $f(x^0)$ is lexicografically maximal over $x \in X$

MMF: $f(x^0)$ is maximal in the MMF sense over $x \in X$

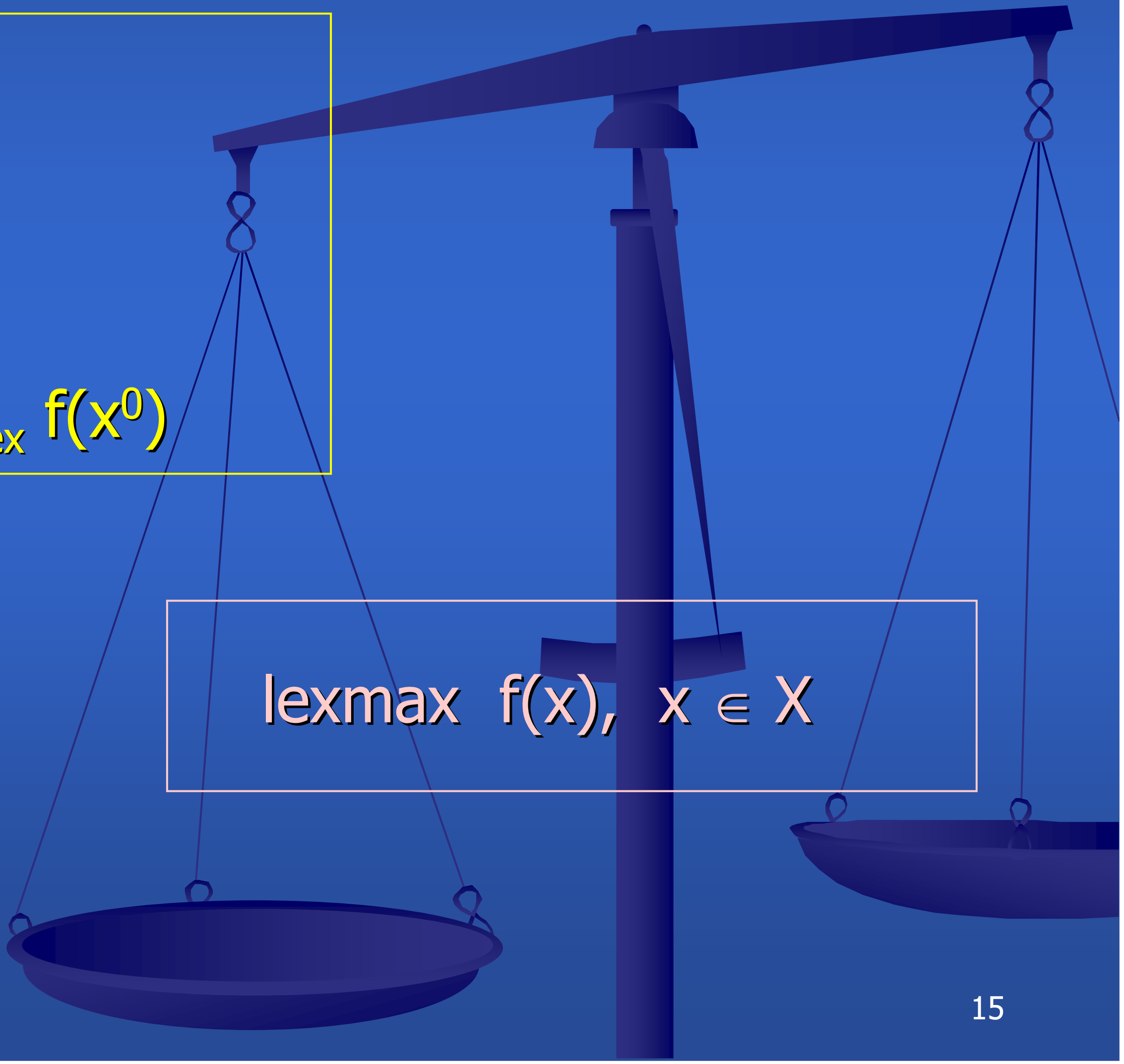
general lexmax problem

Find x^0 lexicographically maximal in X with respect to the criterion function f.

- find $x^0 \in X$
- such that

$$\forall\ x \in X\ ,\ \ f(x) \leq_{lex} f(x^0)$$

$$\text{lexmax}\ \ f(x),\ \ x \in X$$



algorithm for lexmax – steps

Step 0: $k := 1$.

Step 1: Solve the following optimization problem P_k (with $t_1^0, t_2^0, \ldots, t_{k-1}^0$ fixed)

$$\max \{ f_k(x),\ x \in X,\ t_j^0 = f_j(x),\ j=1,2,\ldots,k-1 \}.$$

Denote the resulting optimal solution by x^0 and put $t_k^0 = f(x^0)$.

Step 2: If $k = m$ then stop (x^0 is the optimal solution);
Otherwise put $k := k+1$ and go to Step 1.

Remark: If X is convex and $f_j(x)$ are concave then each P_k is a convex problem.

general MMF problem

- Find $x^0 \in X$
- such that $\forall\ x \in X$, $[\ f(x)\ ] \leq_{lex} [\ f(x^0)\ ]$.

  (find x^0 lexicographically maximal in X
  with respect to the sorted criterion function f)

- The problem is called convex when X is convex and all f_j are concave.
- Convex MMF problems can be treated sequentially in a way that is not much more complex than for lexmax.
- For non-convex problems the procedure is more complex.

MMF algorithm for convex problems – notation

- $M = \{1,2,...,m\}$ index set of the criteria
- $B \subseteq M$ set for which the optimal criteria are already computed
- $t^B = (t_j^B : j \in B)$ vector of optimal criteria (with $|B|$ elements)
- $B' = M \setminus B$ set for which the optimal criteria are to be compted

Problem $P(B,t^B)$ convex

- variables: x, t
- constants: B, t^B

maximize t

subject to

- $f_j(\mathbf{x}) \geq t$ $j \in \mathbf{B'}$ (criteria that can be increased)
- $f_j(\mathbf{x}) = t_j^{\mathbf{B}}$ $j \in \mathbf{B}$ (these criteria are fixed)
- $x \in X$, $t \in R$.

algorithm – steps

Step 0: $B := \emptyset$ and $t^B := \emptyset$.

Step 1: If $B = M$, then STOP (x^0 and $[t^B] = [f(x^0)]$ are optimal).
Else, solve problem $P(B,t^B)$ and denote the resulting solution by (x^0,t^0).
(note that $P(\emptyset,\emptyset)$ yields the first value in the final solution $[f(x^0)]$)

Step 2: For each index $k \in B'$ such that $f_k(x^0) = t^0$ solve the lifting test $T(B,t^B,t^0,k)$:

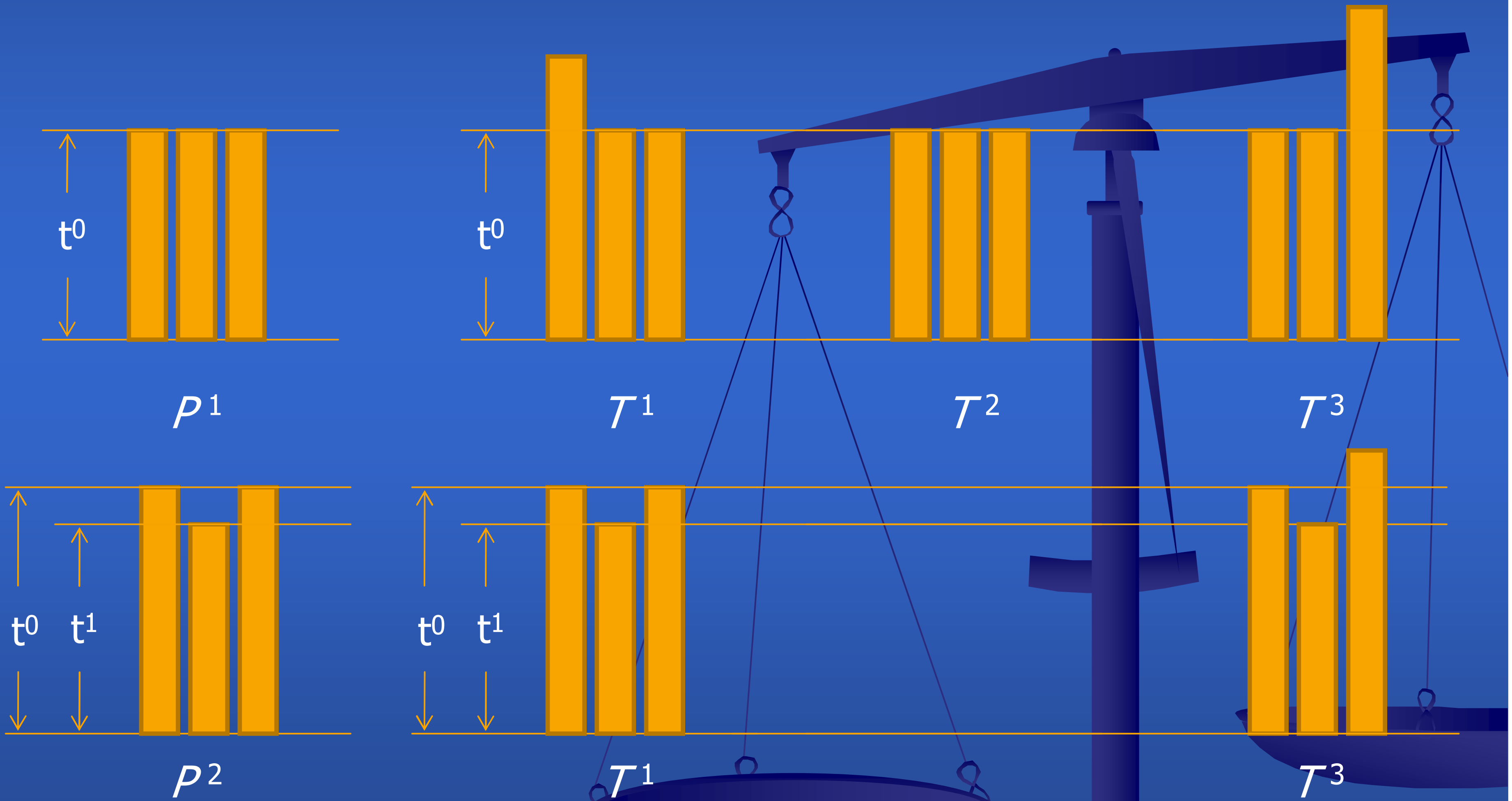
> maximize $f_k(x)$
> subject to
> - $f_j(x) \geq t^0 \qquad j \in B' \setminus \{k\}$
> - $f_j(x) = t_j^B \qquad j \in B$
> - $x \in X$.

If $f_k(x^0) = t^0$ for optimal x^0 solving $T(B,t^B,t^0,k)$ then $B := B \cup \{k\}$, $t_k^B := t^0$.

Step 3: Go to Step 1.

illustration

- example with 3 criteria to be lifted in a MMF way



remarks

- the algorithm works due to convexity: if each criterion can be lifted individually then they all can be lifted simultaneously!
- this property is not present in non-convex problems and this makes them difficult

- excessive number of tests $T(B,t^B,t^0,k)$ to be solved
- several ways to effectively overcome this difficulty
  - use of dual variables
  - making one modified test in each step

formulation of the convex routing problem

Given

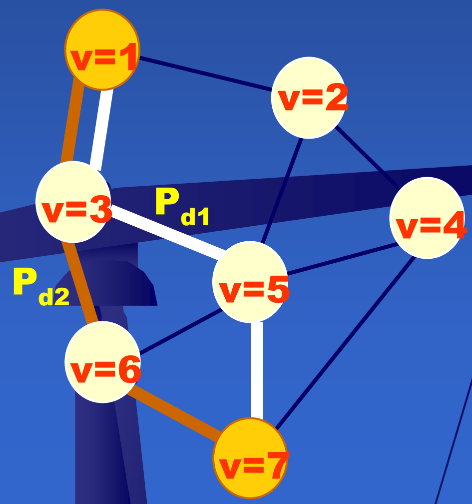
- link capacities (C_e)
- lists of allowable paths
  for realizing demands (P_{d1} , P_{d2} ,..., $P_{dm(d)}$)

Find

- flows assigned to the demand paths (x_{dp})
  (now paths are subject to optimization)

Such that

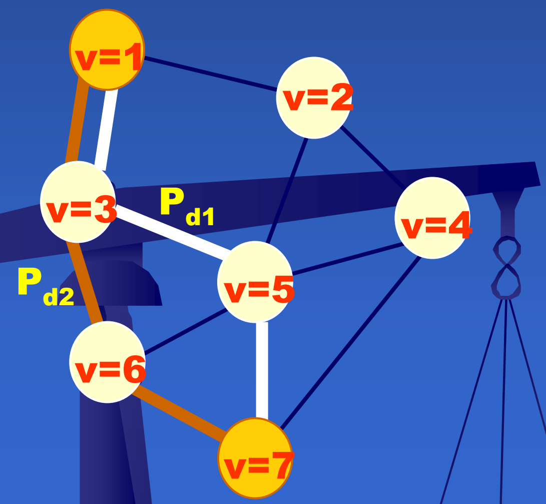
- loads of the links do not exceed their capacity
- flows are assigned to demands in a MMF way:



to maximize lexicographically the sorted total allocation vector $X = (X_1, X_2, \ldots, X_D)$

MMF convex routing problem

- lexmax [$X_1, X_2, \ldots, X_D$]
  - $X_d = x_{d1} + x_{d2} + \ldots + x_{dm(d)}$, $d = 1,2,\ldots,D$
  - $\sum_d \sum_p \delta_{edp} x_{dp} \leq C_e$, $e = 1,2,\ldots,E$
  - variables are continuous and non-negative

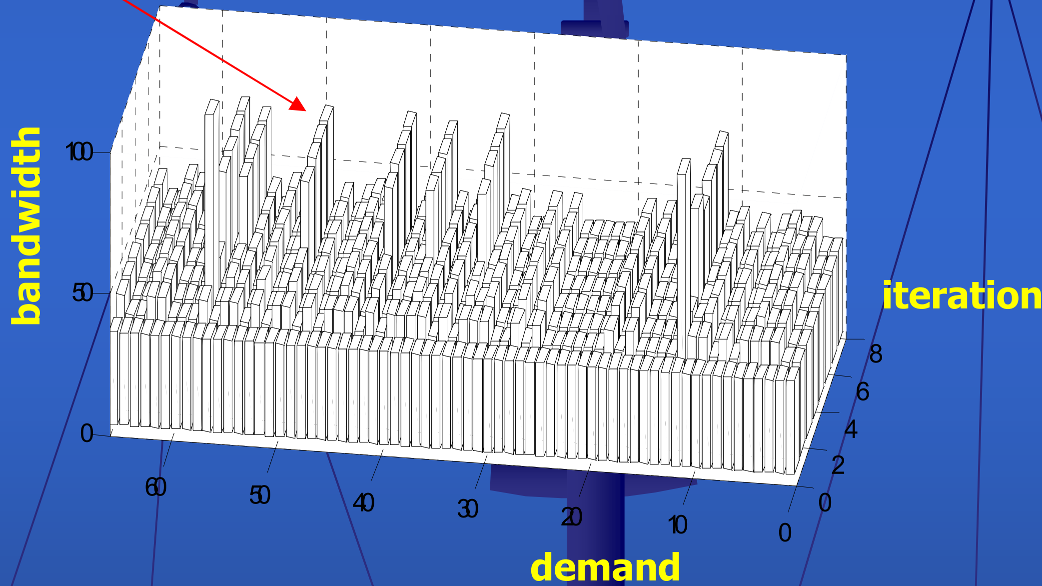


Waterfilling algorithms does not work so we use the general MMF algorithm for convex MMF problems.

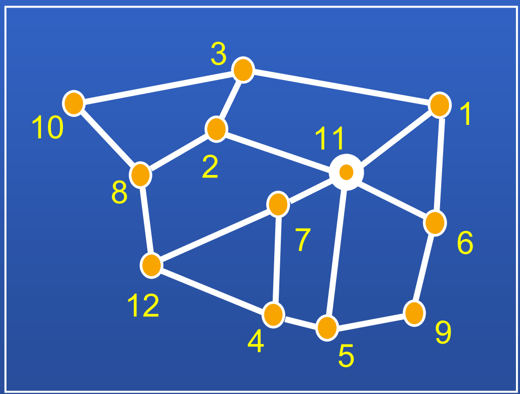
$\delta_{edp} = 1$ if $e \in P_{dp}$

numerical results: 8 steps of the algorithm

optimal total allocation vector $X = (X_1, X_2, \ldots, X_D)$



V=12 nodes
E=18 links
D=66 demands



informal formulation of the routing problem with single-paths (non-convex, *NP* - complete)

- select exactly one path j(d) for each demand d
- allocate entire flow X_d to path $P_{dj(d)}$ (out of $P_{d1}, P_{d2}, \dots, P_{dm(d)}$)
- so that the capacity C_e of no link e is exceeded
- and the vector $[\,(X_1, X_2, \dots, X_D)\,]$ is lexicographically maximal

Remarks

- as we already know when paths j(d) are given and fixed, the problem is easy (waterfilling)
- the difficulty of the problem lies in the path selection

non-convex formulation of the problem (MIP)

Constants

- link capacities (C_e)
- Δ (large enough constant)
- lists of allowable paths for realizing demands (P_{d1} , P_{d2} ,..., $P_{dm(d)}$)

Variables

- flows assigned to the demands paths (x_{dp})
- binary variables associated with flows (u_{dp})

Such that

- $x_{dp} \leq \Delta u_{dp}$ d=1,2,..,D, p=1,2,...,m(d)
- $\sum_p u_{dp} = 1$ d=1,2,...,D
- $\sum_d \sum_p \delta_{edp} x_{dp} \leq C_e$ e=1,2,...,E
- total flows ($X_d = x_{d1} + x_{d2} + \ldots + x_{dm(d)}$) are assigned to demands in the MMF way:

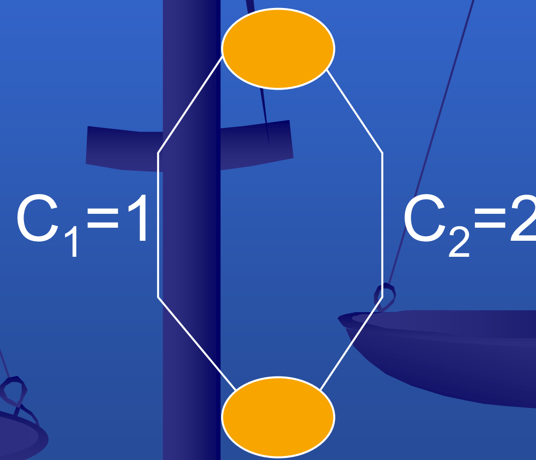
> to maximize lexicographically the sorted total allocation vector $X = (X_1, X_2, \ldots, X_D)$

example

Previous algorithms fail for non-convex X.

Example: two demands between two nodes with two paths of capacity 1 and 2, respectively.

When we solve for the first MMF element we get $(X_1,X_2) = (1,1)$.
The blocking tests will indicate that both criteria can be improved.
They cannot, however, be improved simultaneously.



optimal solution:
$(X_1,X_2) = (1,2)$ or $(X_1,X_2) = (2,1)$

(in the bifurcated case: (1.5,1.5))

transformation of the general problem to linear objective

- $X \subseteq R^n$ a set in n-dimensional Euclidian space
- $x = (x_1, x_2, \ldots, x_n)$ n-vector
- $f(x) = (f_1(x), f_2(x), \ldots, f_m(x))$ $f_j : X \rightarrow R$ scalar functions

lexmax $[f(x)]$ for $x \in X$

- $y = (y_1, y_2, \ldots, y_m)$ m-vector
- $Z \subseteq R^{m+n}$: $(y,x) \in Z$ iff
  - $x \in X$
  - $y_j \leq f_j(x)$, $j=1,2,\ldots,m$

lexmax $[y]$ for $(y,x) \in Z$

optimal x^0 are the same in both problems and $y^0 = f(x^0)$

cumulated criteria

- $[y] = r$
- $R_k = \sum_{j=1,2,..,k} r_j$ $\quad\quad\quad$ $k=1,2,...,m$

lexmax r over Z	is equivalent to	lexmax R over Z

cumulated criteria - derivation of the solution

- $[y] = r$
- $R_k = \sum_{j=1,2,..,k} r_j$ $\quad\quad$ $k=1,2,...,m$

lexmax R over Z

R_k can be expressed as follows (for a fixed k):

$$R_k = \min \sum_j y_j u_{kj}$$

subject to

$$\sum_j u_{kj} = k$$

$$0 \le u_{kj} \le 1 \quad\quad j = 1,2,...,m$$

continuous variables: u_{kj}, $j=1,2,...,k$

LP for fixed y, but non-linear for variable y.

cumulated criteria - derivation of the solution

LP for fixed y, but non-linear for variable y.

But taking the dual gives an LP (for a fixed k):

$R_k =$ **max** $kr_k - \sum_j d_{kj}$

subject to

$d_{kj} \geq r_k - y_j \qquad j = 1,2,\ldots,m$

$d_{kj} \geq 0 \qquad j = 1,2,\ldots,m$

continuous variables: r_k

$d_{kj} \qquad j=1,2,\ldots,k$

cumulated criteria - solution

- $[y] = r$
- $R_k = \sum_{j=1,2,..,k} r_j$ $\qquad$ $k=1,2,...,m$

lexmax R over Z

lexmax R over Z can be expressed as follows:

$$\text{lexmax } (r_1 - \sum_j d_{1j} \,,\, 2r_2 - \sum_j d_{2j} \,,\, \ldots \,,\, mr_m - \sum_j d_{mj})$$

subject to $(y,x) \in Z$

$$d_{kj} \geq r_k - y_j \qquad j, k = 1,2,...,m$$
$$d_{kj} \geq 0 \qquad j, k = 1,2,...,m$$

Can be solved sequentially, for each k=1,2,...,m.

Sequential algorithm – steps

Step 0: $k := 1$.

Step 1: Solve the program

P_k :	max	$kr_k - \sum_j d_{kj}$	
	subject to	$(y,x) \in Z$	
		$R_i^0 \leq ir_i - \sum_j d_{ij}$	$i=1,2,\ldots,k-1$
		$d_{ij} \geq r_i - y_j$	$j=1,2,\ldots,m,\ i=1,2,\ldots,k$
		$d_{ij} \geq 0$	$j=1,2,\ldots,m,\ i=1,2,\ldots,k$

and denote its optimal solution by (x^0,y^0,R_k^0).

Step 2: If $k = m$ then stop (x^0,y^0 is the optimal solution). Otherwise put $k := k+1$ and go to Step 1.

Computation times [sec] for single path allocation

Cumulated criteria
Direct approach based on explicit formulations (for each step of the sequential process)

Nilsson, P.: *Fairness in communication and computer network design*, PhD thesis, Lund University, 2006.

#nodes	#links	#paths	direct	cumulated
5	8	2	0.81	0.47
6	12	2	1.12	1.37
7	12	2	10.1	4.71
8	13	2	16.4	21.1
9	18	2	1622	328
10	18	2	1613	327
11	19	2	1920	107
4	6	3	0.42	0.10
6	12	3	1.15	13.1
7	17	3	26.7	29.0

Remarks

- **Cumulated approach is a general approach to resolving non-convex MMF problems**
- **Cumulated approach adds no difficulty to the resolution scheme with respect to one-criterion versions of the original problem**
- **Cumulated approach performs better than the direct approach**
  - **this suggests effectiveness of the cumulated approach in the general case**

Other important problems involving MMF

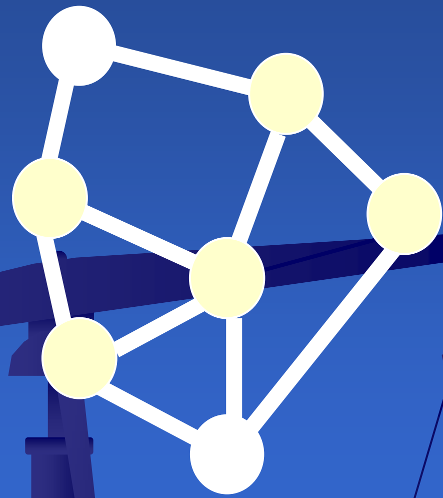
- Maximization of unused capacity
- Introducing resilience to failures in non-protected networks
- Dimensioning of resilient networks

maximization of unused capacity

max Y

- $x_{d1} + x_{d2} + \ldots + x_{dm(d)} = h_d$, $d = 1,2,\ldots,D$
- $\sum_d \sum_p \delta_{edp} x_{dp} + Y \le C_e$, $e = 1,2,\ldots,E$
- variables are continuous and non-negative

This is the first step of the MMF problem given below. Many people considered that first step and did not know how to continue!



correct MMF formulation:

- lexmax [$Y_1, Y_2, \ldots, Y_E$]
  - $x_{d1} + x_{d2} + \ldots + x_{dm(d)} = h_d$, $d = 1,2,\ldots,D$
  - $\sum_d \sum_p \delta_{edp} x_{dp} + Y_e \le C_e$, $e = 1,2,\ldots,E$
  - variables are continuous and non-negative

Conclusions

- **MMF is useful in network design**
  - **routing problems (for elastic traffic)**
  - **unsed capacity maximization**
  - **protection problem**
  - **many others**
- **Convex MMF problems can be effectively solved through a sequential procedure involving a master problem and lifting tests**
- **There is a way to effectively incorporate MMF into non-convex problems, without increasing their complexity (only continuous variables and linear constraints added)**
  - **e.g., bandwidth allocation problem involving single paths**

more in:

- D. Nace, M. Pioro: Max-min fairness and its applications to routing and load-balancing in communication networks – a tutorial, *IEEE Communications Surveys and Tutorials*, vol.10, no.4, pp.5-17, 2008
- W. Ogryczak, M. Pioro, A. Tomaszewski: Telecommunications network design and max-min optimization problem, *Journal of Telecommunications and Information Technology*, No.3, 2005
- M. Pioro, D. Medhi: *Routing, flow, and capacity design in communication and computer networks (chapters 8 and 13)*, Morgan-Kaufmann (Elsevier), 2004

Thank you!

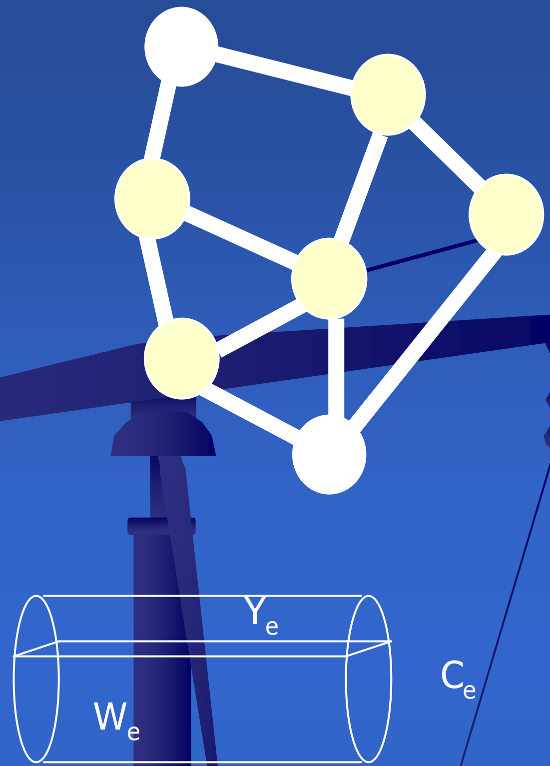
protection of a network

Given:

- link capacities $C_1, C_2, \ldots, C_E$
- realized demand volumes: $h_1, h_2, \ldots, h_D$

Problem

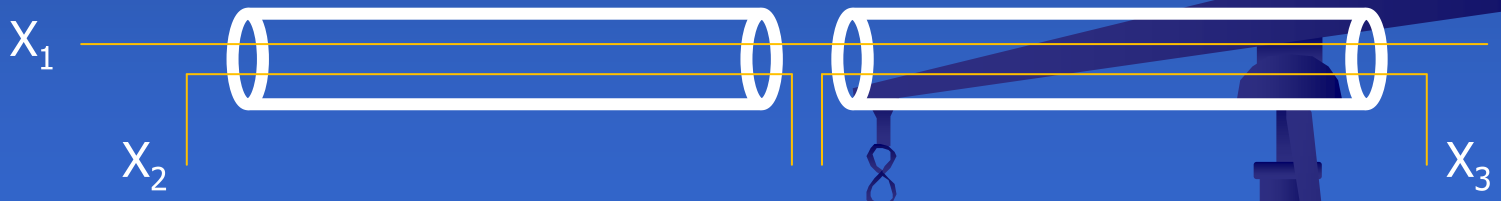
- for each link divide its capacity C_e into working capacity W_e and protection capacity Y_e so that in the case of failure of any single link g
  - its working capacity W_g can be restored using protection capacities Y_e $(e \neq g)$
  - demand volumes $th_1, th_2, \ldots, th_D$ can be realized in working capacities W_e $(e = 1,2,\ldots,E)$
- t is maximized



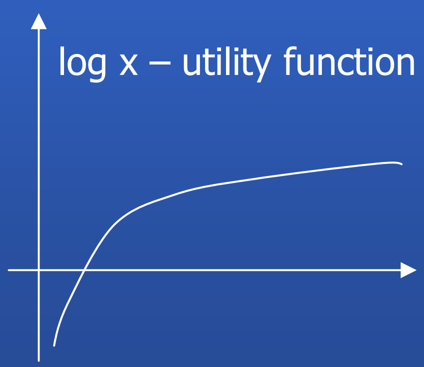
- the above problem is the first step in the MMF problem
  - lexmax $[\, t_1, t_2, \ldots, t_D \,]$
  - demand volumes $t_1h_1, t_2h_2, \ldots, t_Dh_D$ are realized in working capacities W_e $(e = 1,2,\ldots,E)$
  - working capacity W_g of any link g can be restored using protection capacities Y_e $(e \neq g)$

routing problem for a simple network

- two links in series – each of capacity 10 (e.g., 10 Mbps)
- three elastic demands (flows) eager to get as much bandwidth as possible



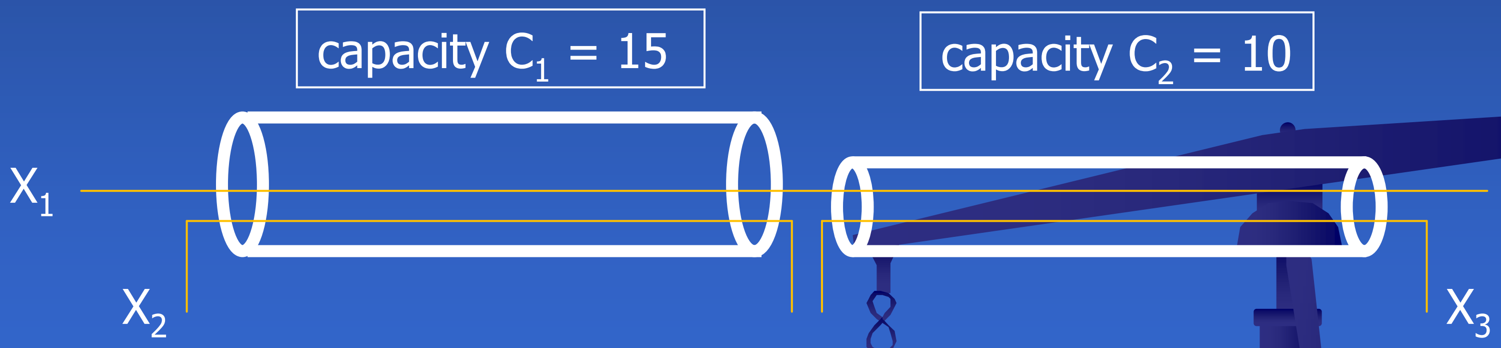
- proportional fairness: $X_1 = 3.33, X_2 = X_3 = 6.66$ (16.66)
  - maximize ($\log X_1 + \log X_2 + \log X_3$) with respect to capacity constraints



throughput

maximize $f(x) = \log x + 2\log(1-x)$

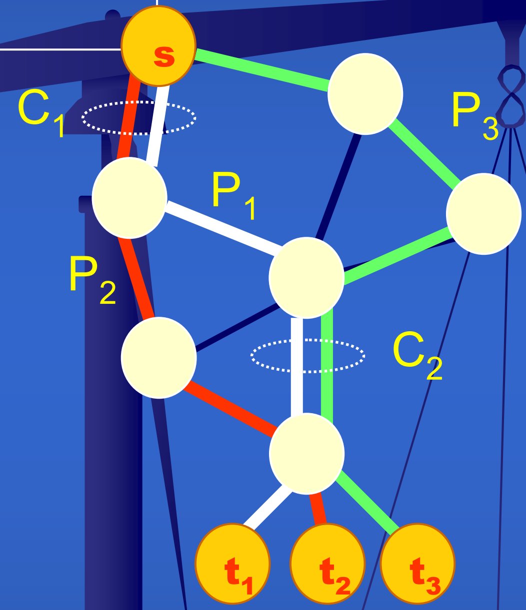
routing problem for a simple network (modified)



- lexicographical fairness (priority 3,2,1): $X_1 = 0$, $X_2 = 15$, $X_3 = 10$ (25)
- max-min fairness: $X_1 = 5$, $X_2 = 10$, $X_3 = 5$ (20)
- proportional fairness: $X_1 \approx 4$, $X_2 \approx 11$, $X_3 \approx 6$ (21)

PF routing – solution

$C_1 = 15$ $C_2 = 10$ (bottleneck links)
the rest of links have large capacity
the amount of goods (bandwdth) is infinite



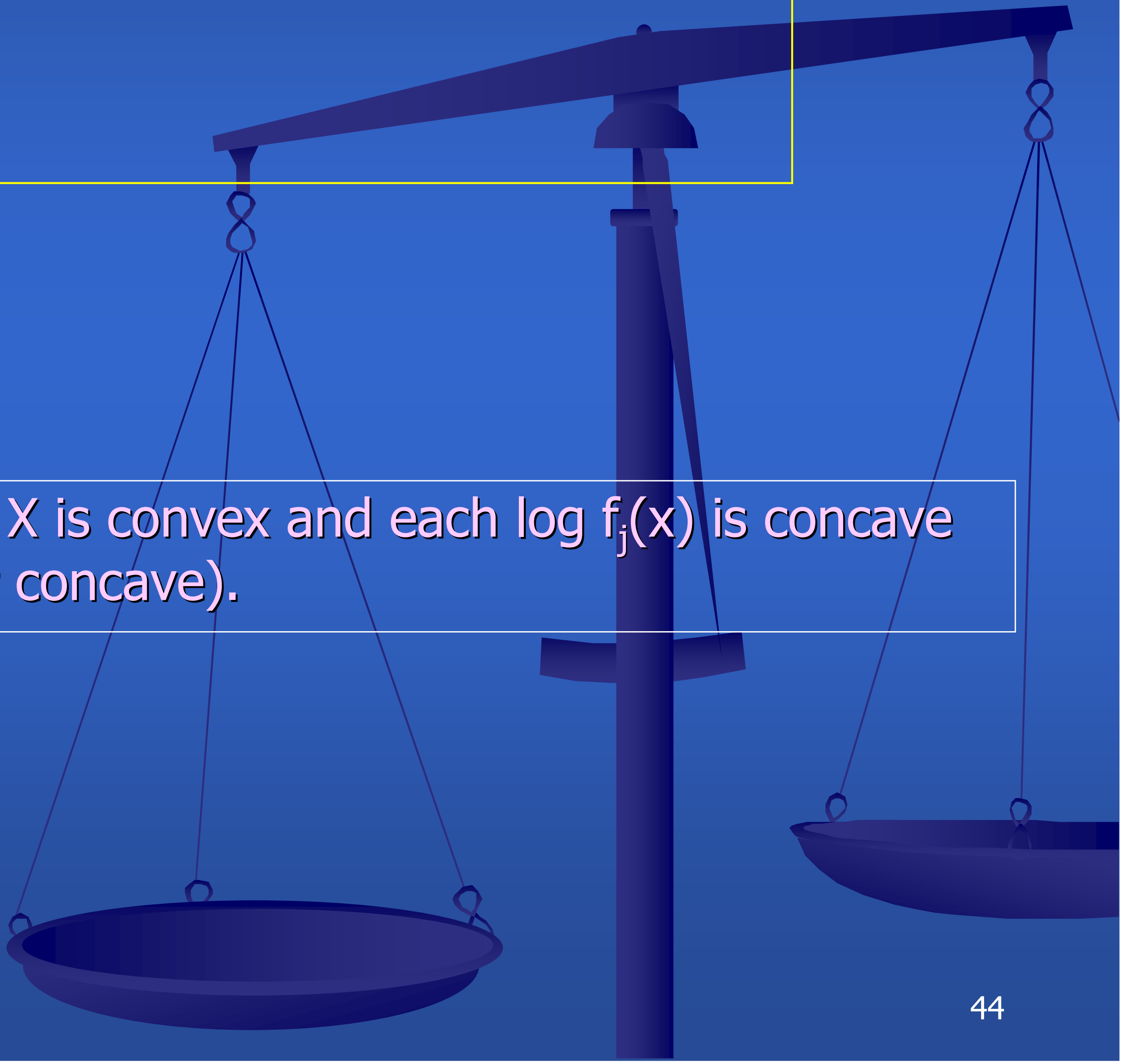
- $X_1 + X_2 \leq 15$
- $X_1 + X_3 \leq 10$
- maximize $\log X_1 + \log X_2 + \log X_3$

solution (standard convex optimization problem):

- $X_1 \approx 4$ $X_2 \approx 11$ $X_3 \approx 6$

general PF problem

- maximize $\log f_1(x) + \log f_2(x) + \ldots + \log f_m(x)$
- over $x \in X$



The problem is convex when X is convex and each $\log f_j(x)$ is concave (e.g., when $f_j(x)$ are linear or concave).

utility function

- $U(X) = r\, X^{(1-\alpha)} / (1 - \alpha)$
  - throughput maximization: $\alpha = 0$, $U(X) = r\, X$
  - MMF: $\alpha \rightarrow \infty$
  - PF: $\alpha \rightarrow 1$, $U(X) = r \log X$