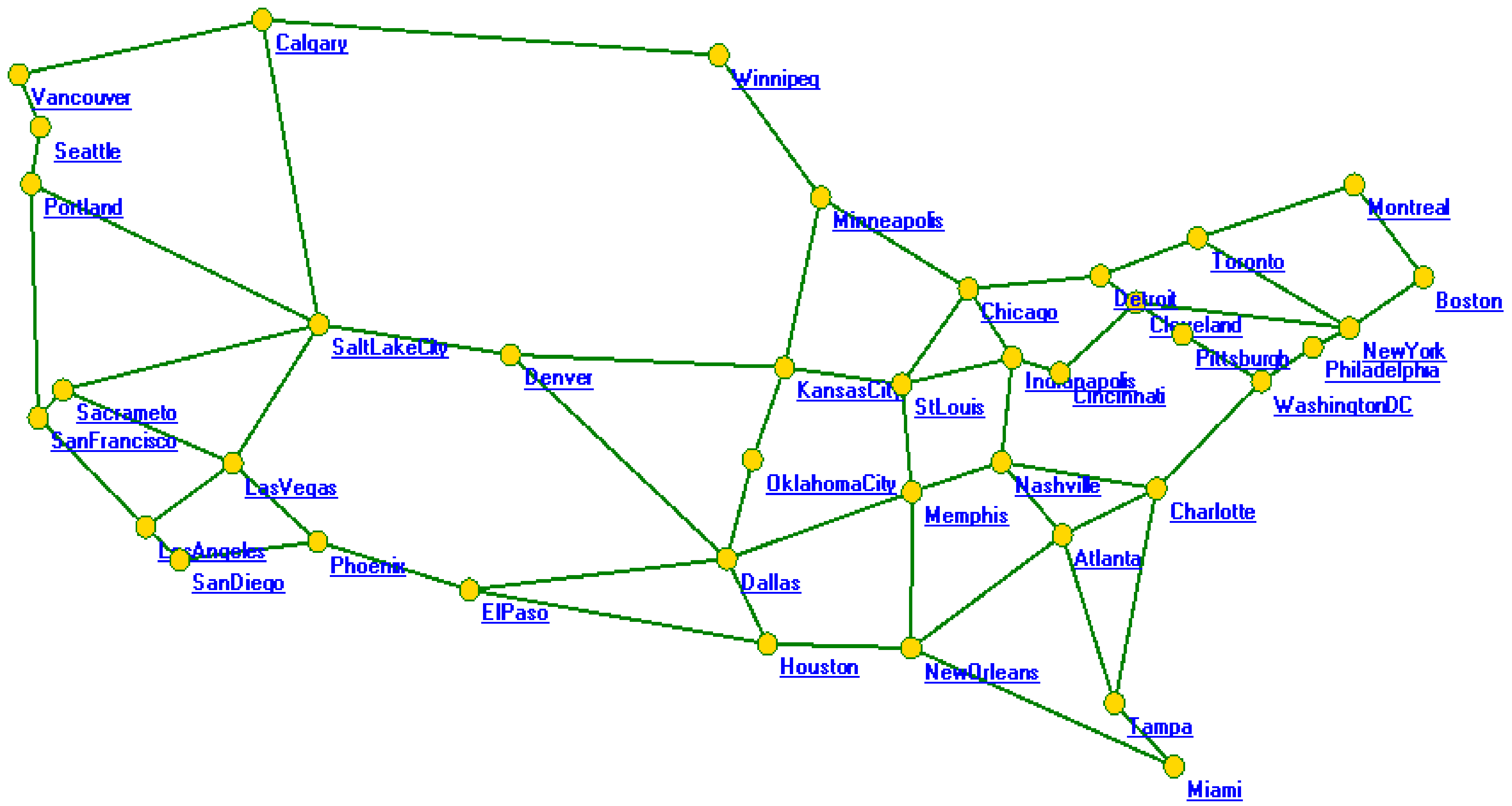
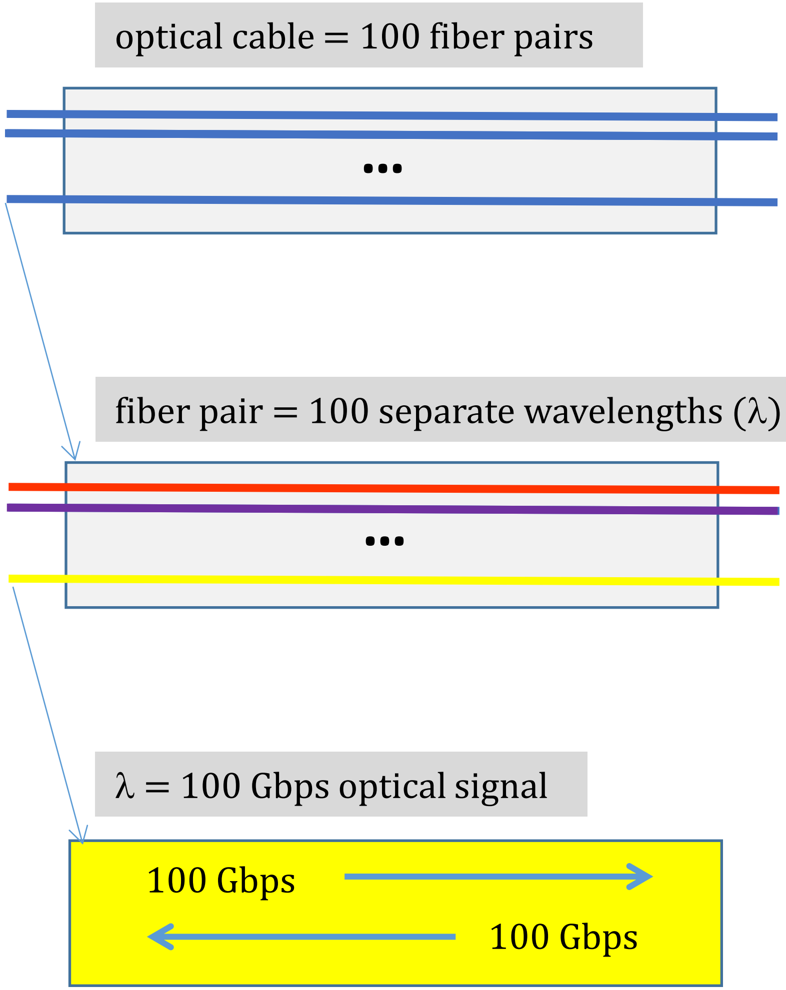
backbone optical network 1

sndlib.zib.de



Dense Wavelength Division Multiplexing (DWDM) transmission systems



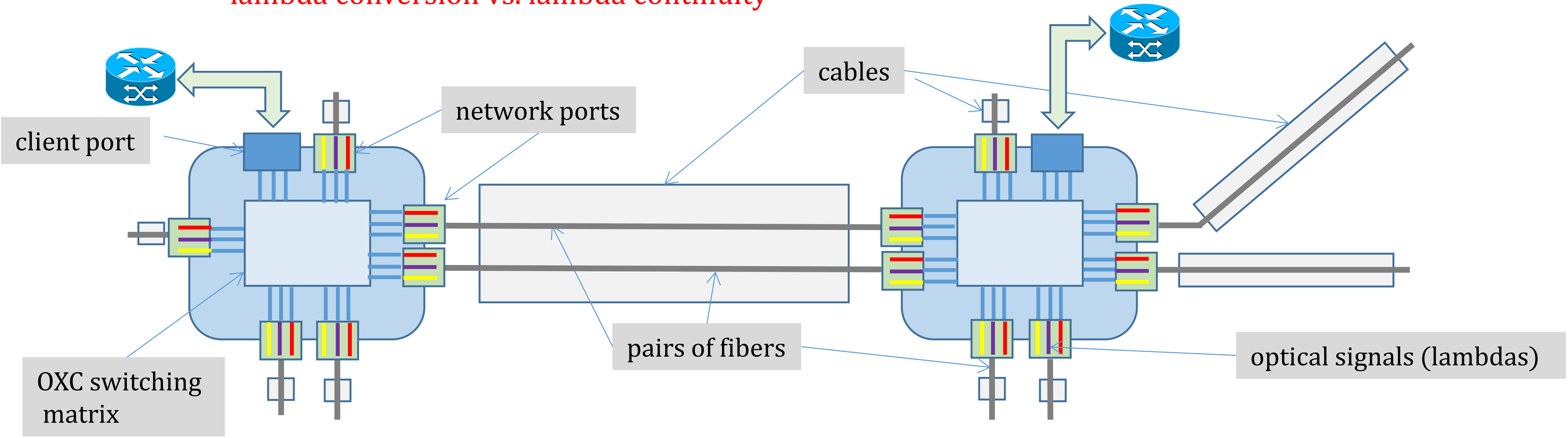
DWDM transmission system = one fiber pair

in summary:

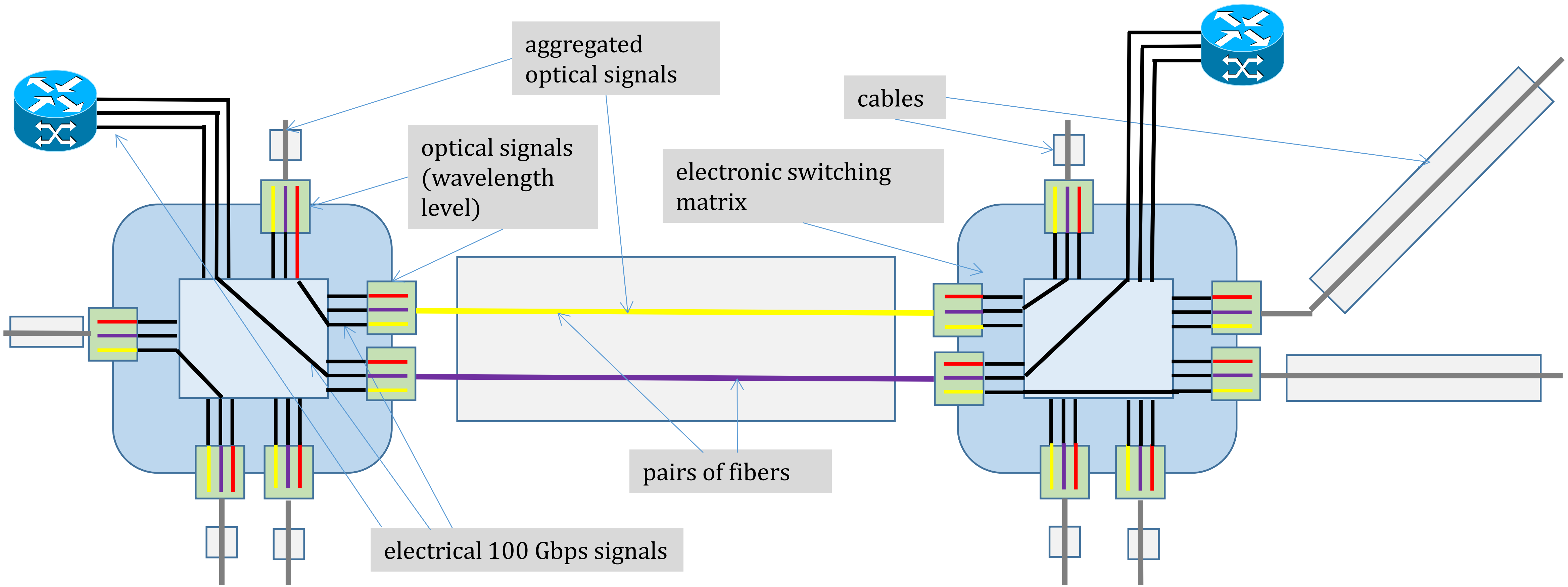
- one DWDM transmission system (fiber pair) carries 100×100 Gbps = 10 000 Gbps = 10 Tbps in each direction
- one cable can carry 100×10 Tbps = 1000 Tbps = 1 Pbps

DWDM backbone networks

- **DWDM transmission system (full duplex)**
  - realized on a pair of fibers terminated at a port at each of the end nodes of the supporting link
  - carries 100 optical data streams, each using a separate wavelength, called lambda (λ) or color
    - each lambda = 100 Gbps $\Rightarrow$ DWDM transmission system on a fiber pair = 10 Tbps in each direction
    - optical cable = 100 pairs of fibers $\Rightarrow$ capacity of a cable = 1000 Tbps in each direction
- **DWDM switching node** – optical cross connects (**OXC**)
  - composed of ports and a switching matrix (plus control)
  - port: terminates one DWDM transmission system
  - ports and the switching matrix connected on the lambda signal level
  - switching matrix can connect any two lambda optical signals from any two different ports
    - lambda conversion vs. lambda continuity

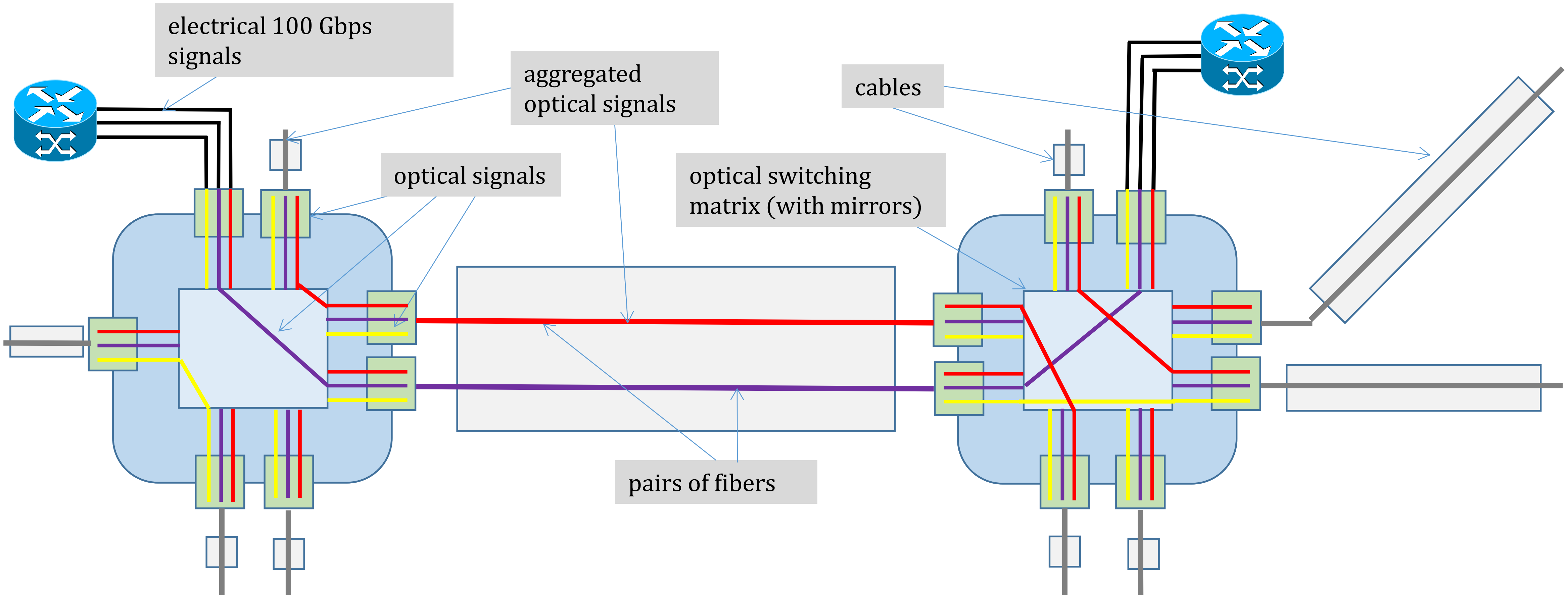


DWDM backbone network (wavelength conversion)



- optical-cross-connect (switches)
  - optical-to-electrical signal conversion and electronic switching

DWDM backbone network (wavelength continuity)



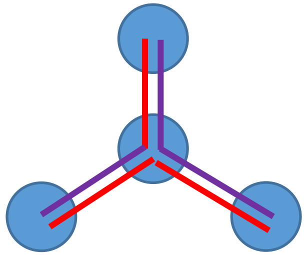
- MEMS: micro-electro-mechanical-system
  - a switch that reflects light beams (of a fixed wavelength) using mirrors; lambdas extracted/combined by means of optical filters
  - https://www.slideshare.net/hi2mohdnazir/mems-optical-switches

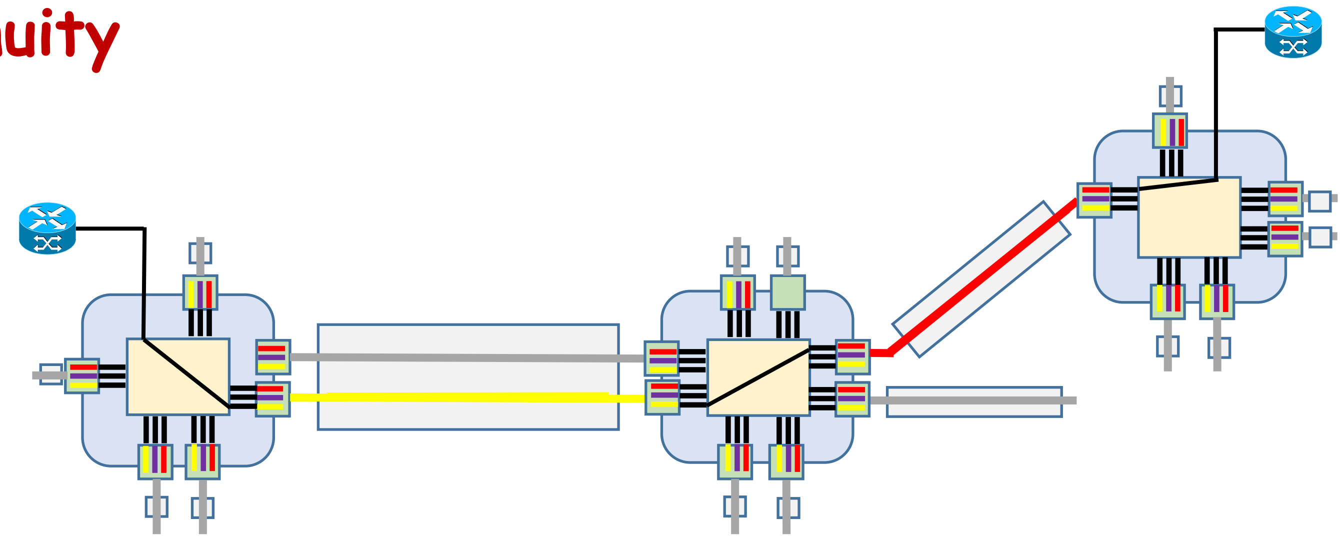
light-paths (λ-paths)

- consider route P (a simple path in the network graph) connecting two nodes o and t
- route P supports pairs of oppositely directed optical connections established by means of λ-switching at the OXC:s along the route
- optical connection realizes an end-to-end optical signal over a sequence of fiber pairs with a (fiber-specific or fixed) wavelength assigned to the signal on each fiber pair
- such connections are called light-paths (or lambda-paths = λ-paths)
- light-path is terminated at a client port in nodes o and t
- capacity of a light-path is equal to 100 Gbps in each direction
- thus, one light-path connecting o and t can carry data at rate 100 Gbps, realizing packet traffic generated at the routers connected to o and t
- one fiber pair on a link can carry up to 100 λ-paths, i.e., 10 Tbps
- one optical cable contains up to 100 pairs of fibers
- one link (=optical cable) can thus support $100 \times 100 = 10\,000$ λ-paths, i.e., 1000 Tbps in each direction

lambda conversion vs. lambda continuity

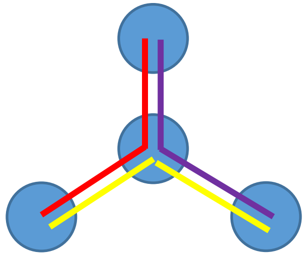
- λ-path with lambda conversion:
  - optical-electrical signal conversion at the nodes
  - less capacity (assuming $\Lambda = 2$, one fiber is sufficient)

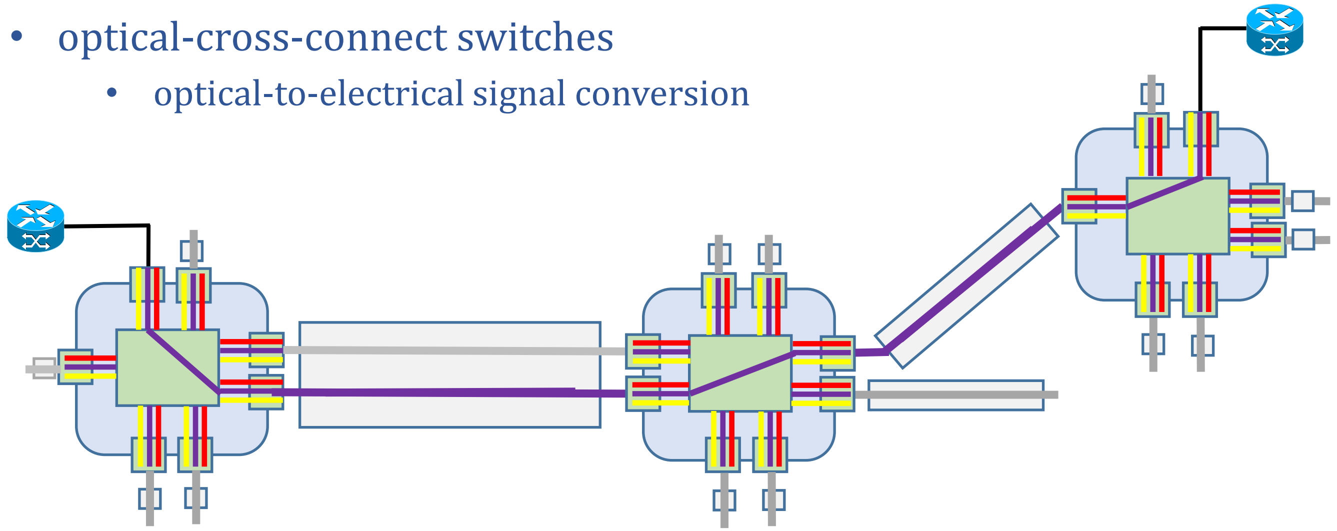




- optical-cross-connect switches
  - optical-to-electrical signal conversion

- λ-path with lambda continuity:
  - no optical-electrical signal conversion = cheaper switching devices
  - more capacity (assuming $\Lambda = 2$, two fibers needed)

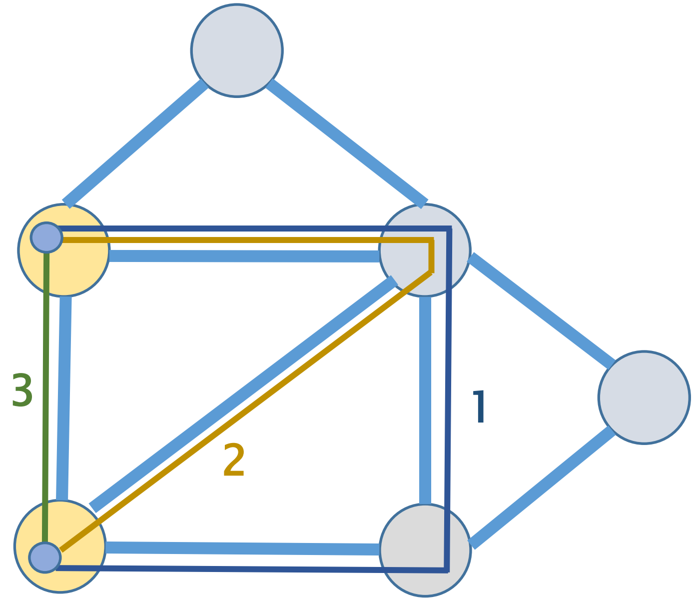




- MEMS: micro-electro-mechanical-system
  - a switch that reflects light beams (of a fixed wavelength) using mirrors (lambdas extracted/combined by means of optical filters)
  - https://www.slideshare.net/hi2mohdnazir/mems-optical-switch

routing of light-paths and the resulting link loads (wavelength conversion)

- given undirected network graph
- given undirected demands:
  - volumes h(d) (in light-paths), routes $\mathcal{P}(d,1)$, $\mathcal{P}(d,2)$, ... ,$\mathcal{P}(d,P_d)$
- given M (M= 100 – number of lambdas per fiber)



solution x defines light-paths' allocation (routing)
x(d',p) = number of light-paths realized on route number p of demand d'

demands d=1,2,...,D

routes p=1,2,...,Pd

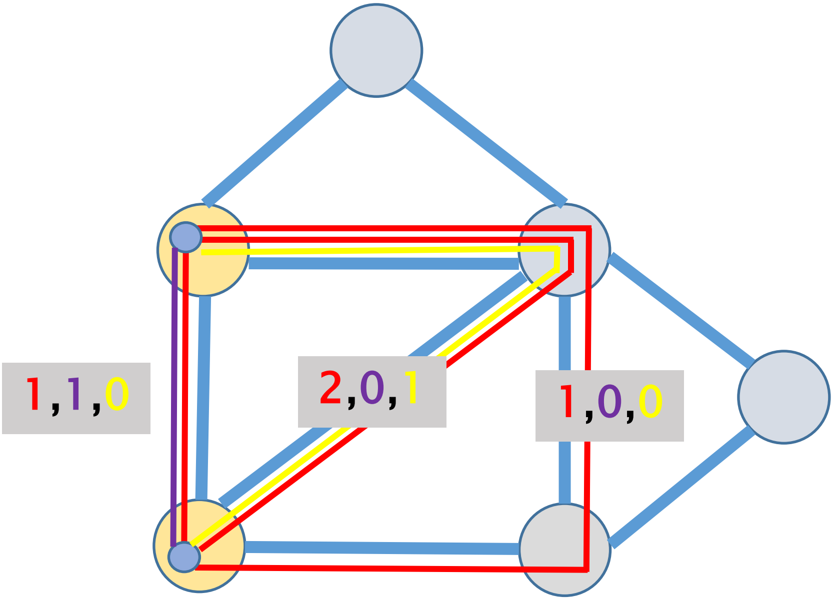
	d=1	d=2		d=D
	0	3		1
	3	2		3
	4	1		0
	0			1
	1			
	h(1)=8	h(2)=6		h(D)=5

- find link load l(e,x) ($e \in \mathcal{E}$):
  number of required lambdas =
  number of light-paths using link e

- then compute link capacity y(e,x):
  i.e., the number of fiber pairs required to carry l(e,x):
  $y(e,x) = \lceil l(e,x)/M \rceil$

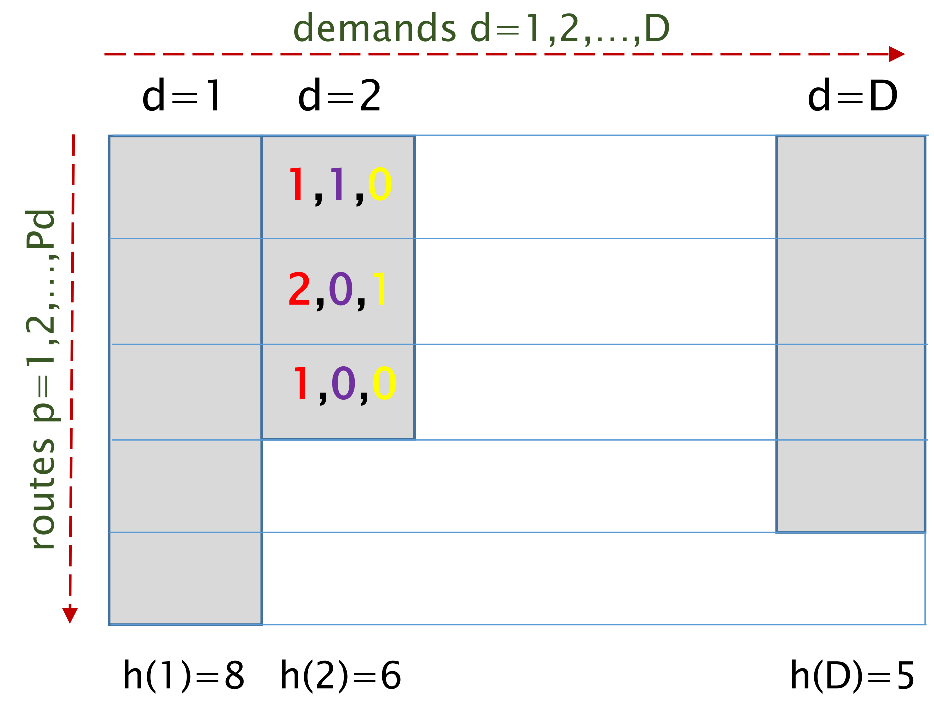
routing of light-paths with wavelength continuity

- volumes h(d), routes $\mathcal{P}(d,1), \mathcal{P}(d,2), \dots, \mathcal{P}(d,P_d)$
- a colour must be assigned to each λ-path realized on a route
- a new dimension (λ)



solution x defines λ-path allocation (routing) x(d,p,λ) = number of lambda-paths of colour λ realized on route number p of demand d

x(d,p,λ):



- find l(e,λ,x) ($e \in \mathcal{E}$, λ=1,2,…,M): number of λ-paths using wavelength λ traversing link e
- then compute $y(e,x)=\max\{\, l(e,\lambda,x): \lambda=1,2,\dots,M \,\}$: number of fiber pairs required to carry the λ-paths on link e
- efficiency of EA? mutations?

DDAP – integer programming (IP) formulation

Book: Section 4.3.1

- parameters M, ξ_e, h_d — M – module size, ξ_e – cost of installing one module on link e, h_d –volume of demand d
- variables x_{dp}, y_e — x_{dp} – path-flow on route $\mathcal{P}_{dp}$, y_e – number of modules on link e

- minimize $F = \sum_{e=1}^{E} \xi_e y_e$
- subject to

$$\sum_{p=1}^{P_d} x_{dp} = h_d \qquad d = 1,2,\dots,D$$

$$\sum_{d=1}^{D} \sum_{p=1}^{P_d} \delta_{edp} x_{dp} \le M y_e \qquad e = 1,2,\dots,E$$

$$x_{dp} \in Z_+ \qquad d = 1,2,\dots,D, \ \ p = 1,2,\dots,P_d$$

$$y_e \in Z_+ \qquad e = 1,2,\dots,E$$

Z_+ – set on non-negative integer numbers

remark: range of summation will be skipped if full

how to solve DDAP (and DAP) exactly?

- general method for integer programming problems: branch-and-bound (B&B)

- the B&B algorithm makes use of linear relaxation (LR) of the problem in hand
  - LR provides a lower bound for the integer (true) version of DDAP
  - in addition, LR gives a suboptimal solution (by rounding fractional solution), which provides an upper bound for the exact integer solution

- LR is a linear programming problem
  - general method for linear programming problems: the simplex algorithm

DDAP – linear programming (LP) formulation (linear relaxation of IP)

Book: Section 2.4

- variables $x_{dp}, y_e{}'$ x_{dp} - path-flow on route $\mathcal{P}_{dp}$

- minimize $F = \sum_e \xi_e{}' y_e{}'$
- subject to $\sum_p x_{dp} = h_d$ $d = 1,2,\dots,D$

$\sum_d \sum_p \delta_{edp} x_{dp} \le y_e{}'$ $e = 1,2,\dots,E$

$x_{dp} \in R_+$ $d = 1,2,\dots,D,\ p = 1,2,\dots,P_d$

$y_e{}' \in R_+$ $e = 1,2,\dots,E$

R_+ – set on non-negative real numbers

remark 1: where $y_e{}'$ is used instead of My_e and $\xi_e' = \xi_e/M$ instead of ξ_e

remark 2: optimal objective $F^* = F^*((y')^*)$ is a lower bound for the integer problem

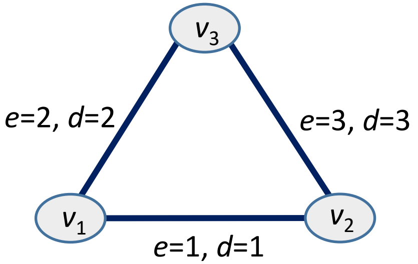
remark 3: by rounding off optimal flows $x^*$and then finding the resulting link sizes y we obtain a reasonable feasible solution for the integer version of DDAP; the latter provides an upper bound on the true (i.e., integer) solution

remark 4: evolutionary algorithm cannot be applied in an efficient way (granularity)

example: DDAP

$$E = 3, D = 3$$
$$M = 2,\ \xi_e \equiv 1, h_d \equiv 1$$

$$\mathcal{P}_{11} = \{1\}, \mathcal{P}_{12} = \{2,3\}$$
$$\mathcal{P}_{21} = \{2\}, \mathcal{P}_{22} = \{1,3\}$$
$$\mathcal{P}_{31} = \{3\}, \mathcal{P}_{32} = \{1,2\}$$



$$\min\ \sum_e \xi_e y_e$$
$$\sum_p x_{dp} = h_d \qquad d = 1,2,\dots,D$$
$$\sum_d \sum_p \delta_{edp} x_{dp} \le M y_e \qquad e = 1,2,\dots,E$$

$$\min\ F = \sum_e y_e$$

$$\begin{array}{llllllll}
x_{11} + x_{12} & & & & & = 1 & & (d=1) \\
& x_{21} + x_{22} & & & & = 1 & & (d=2) \\
& & x_{31} + x_{32} & & & = 1 & & (d=3) \\
x_{11} & + x_{22} & & + x_{32} & & \le 2y_1 & & (e=1) \\
x_{12} + x_{21} & & & + x_{32} & & \le 2y_2 & & (e=2) \\
x_{12} & + x_{22} + x_{31} & & & & \le 2y_3 & & (e=3)
\end{array}$$

IP vs. LP solution?

optimal solution for IP:
$x_{12} = 1, x_{21} = 1, x_{32} = 1$ (the rest 0)
$F(x) = 2$ (2 modules suffice)

optimal solution for LP:
$x_{11} = 1, x_{21} = 1, x_{31} = 1$ (the rest 0)
$F(x) = 1.5$
each link costs 0.5 since it requires half of M
which means that $y_e \equiv 0.5$

DAP – linear version (continuous flows)

Book: problem 4.1.8

- parameters $M, c_e, C_e = Mc_e, h_d$
- variables x_{dp} x_{dp} - path-flow on route $\mathcal{P}_{dp}$

- minimize z
- subject to

$$\textstyle\sum_p x_{dp} = h_d \qquad d = 1,2,\dots,D$$

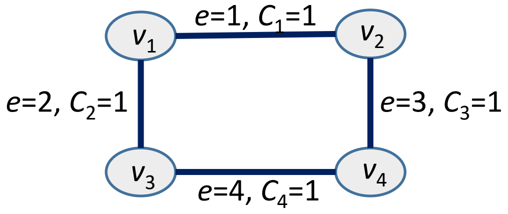
$$\textstyle\sum_d \sum_p \delta_{edp} x_{dp} \le C_e + z \qquad e = 1,2,\dots,E$$

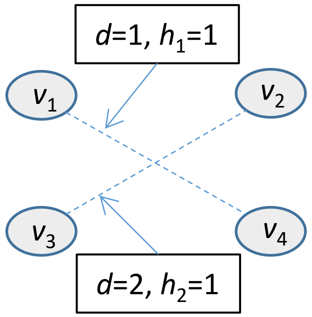
$$x_{dp} \in R_+ \qquad d = 1,2,\dots,D,\;\; p = 1,2,\dots,P_d$$

$$z \in R$$

R_+ – set of non-negative real numbers (R – set of real numbers)

example: DAP





$E = 4, D = 2$

$\mathcal{P}_{11} = \{1,3\}, \mathcal{P}_{12} = \{2,4\}$
$\mathcal{P}_{21} = \{2,1\}, \mathcal{P}_{22} = \{4,3\}$

$$\min\ z$$
$$\sum_p x_{dp} = h_d \qquad d = 1,2,\dots,D$$
$$\sum_d \sum_p \delta_{edp} x_{dp} \le C_e + z \qquad e = 1,2,\dots,E$$

$$
\begin{array}{lr}
\min\ z & \\
x_{11} + x_{12} = 1 & (d = 1) \\
x_{21} + x_{22} = 1 & (d = 2) \\
x_{11} + x_{21} \le 1 + z & (e = 1) \\
x_{12} + x_{21} \le 1 + z & (e = 2) \\
x_{11} + x_{22} \le 1 + z & (e = 3) \\
x_{12} + x_{22} \le 1 + z & (e = 4)
\end{array}
$$

optimal solution for IP:
$x_{11} = 1, x_{12} = 0, x_{21} = 1, x_{22} = 0$
$z = 1$ because load of $e = 1$ is equal to 2

optimal solution for LP:
$x_{11} = 0.5, x_{12} = 0.5, x_{21} = 0.5, x_{22} = 0.5$
$z = 0$ all links saturated but bot overloaded

IP vs. LP solution?

variants of DDAP/DAP - path diversity

Book: problem 4.2.1

- additional parameter n_d – bifurcation coefficient
- variables x_{dp}, y_e

modified notation I

- minimize $F = \sum_{e \in \mathcal{E}} \xi_e y_e$
- subject to

$$\sum_{p \in \mathcal{P}(d)} x_{dp} = h_d \qquad d \in \mathcal{D} \qquad \mathcal{D} = \{1,2, \dots, D\}$$

$$x_{dp} \le \frac{h_d}{n_d} \qquad d \in \mathcal{D},\ p \in \mathcal{P}(d) \qquad \mathcal{P}(d) = \{1,2, \dots, P_d\}$$

$$\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}(d)} \delta_{edp} x_{dp} \le y_e \qquad e \in \mathcal{E} \qquad \mathcal{E} = \{1,2, \dots, E\}$$

$$x_{dp} \in R_+ \qquad d \in \mathcal{D},\ p \in \mathcal{P}(d)$$

$$y_e \in R_+ \qquad e \in \mathcal{E}$$

makes sense for disjoint paths

variants of DDAP/DAP - single-path routing

Book: problem 4.2.4

- variables u_{dp}, x_{dp}, y_e
- minimize $F = \sum_e \xi_e y_e \quad (z)$
- subject to

$\sum_p u_{dp} = 1 \qquad d \in \mathcal{D}$

$x_{dp} = h_d u_{dp} \qquad d \in \mathcal{D},\ p \in \mathcal{P}(d) \qquad$ (x_{dp} – absolute flow, can be omitted)

$\sum_d \sum_p \delta_{edp} x_{dp} \le y_e \quad (C_e + z) \qquad e \in \mathcal{E}$

$\boxed{B = \{0,1\}}$

$u_{dp} \in B \qquad d \in \mathcal{D},\ p \in \mathcal{P}(d)$

$x_{dp} \in R_+ \qquad d \in \mathcal{D},\ p \in \mathcal{P}(d)$

$y_e \in R_+ \qquad e \in \mathcal{E}$

variants of DDAP/DAP – limits on path-flows (between *l*(*d*) and *k*(*d*))

- variables u_{dp}, x_{dp}, y_e

- minimize $F = \sum_{e \in \mathcal{E}} \xi_e y_e$
- subject to $\sum_{p \in \mathcal{P}(d)} x_{dp} = h_d \quad d \in \mathcal{D}$

$l(d) u_{dp} \le x_{dp} \le k(d) u_{dp} \quad d \in \mathcal{D},\ p \in \mathcal{P}(d)$

$\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{Q}(d,e)} x_{dp} \le y_e \quad e \in \mathcal{E}$ $\quad \mathcal{Q}(d,e)$ – subset of paths in $\mathcal{P}(d)$ that contain e

(instead of: $\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}(d)} \delta_{edp} x_{dp} \le y_e$)

modified notation II

$u_{dp} \in B \quad d \in D,\ p \in \mathcal{P}(d)$

$x_{dp} \in R_+ \quad d \in D,\ p \in \mathcal{P}(d)$

$y_e \in R_+ \quad e \in \mathcal{E}$

exactly $r(d)$ paths?

DAP – no more than 4 path-flows between 1 and h(d)/2 (h(d) ≥ 2)

- variables u_{dp}, x_{dp}, y_e
- minimize z
- subject to

$$\sum_{p \in \mathcal{P}(d)} x_{dp} = h_d \qquad d \in \mathcal{D}$$

$$\sum_{p \in \mathcal{P}(d)} u_{dp} \le 4 \qquad d \in \mathcal{D}$$

$$u_{dp} \le x_{dp} \le \left(\frac{h_d}{2}\right) u_{dp} \qquad d \in \mathcal{D},\ p \in \mathcal{P}(d)$$

$$\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{Q}(d,e)} x_{dp} \le C(e) + z \qquad e \in \mathcal{E}$$

$$u_{dp} \in B \qquad d \in \mathcal{D},\ p \in \mathcal{P}(d)$$

$$x_{dp} \in R_+ \qquad d \in \mathcal{D},\ p \in \mathcal{P}(d)$$

$$y_e \in R_+ \qquad e \in \mathcal{E}$$

$$z \in R$$

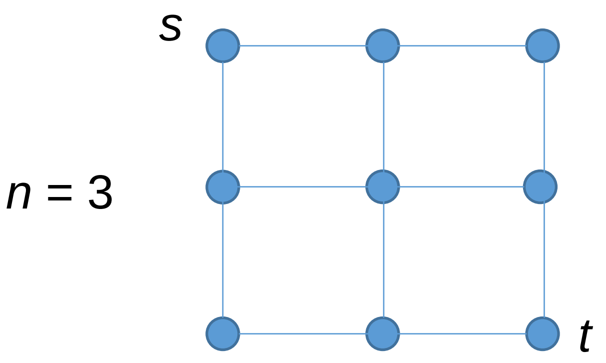
chromosomes?

link-path (L-P) formulations – pros and cons

- link–path formulations
  - uses predefined lists of allowable paths: $\mathcal{P}_{dp},\ d = 1,2,\dots,D,\ p = 1,2,\dots,P_d$
    ($\mathcal{P}_{dp} \subseteq \{1,2,\dots,E\}$ – subset of the set of links indices)
  - $\delta_{edp} = 1$ iff link e belongs to path p realizing demand d; $\delta_{edp} = 0$, otherwise,
    ($\delta_{edp} = 1$ iff $e \in \mathcal{P}_{dp}$)
  - and the corresponding path–flow variables x_{dp}
- pros:
  - more general than alternative formulations
  - limited number of variables when lists are short
- cons:
  - not clear how to predefine proper path–lists (all paths cannot be put on the lists)

number of paths in Manhattan networks

number of paths in the graph grows exponentially so we simply cannot put them all on the path lists!



The number of shortest paths (each shortest path has $2(n-1)$ links) from s to t is equal to

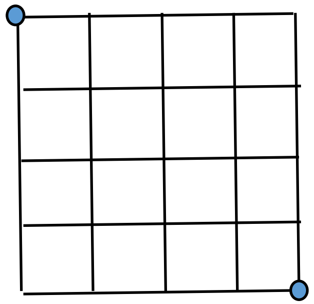
$$\binom{2n-2}{n-1}$$

$> 2^n$

In the above example it is 4 over 2, i.e., 6.

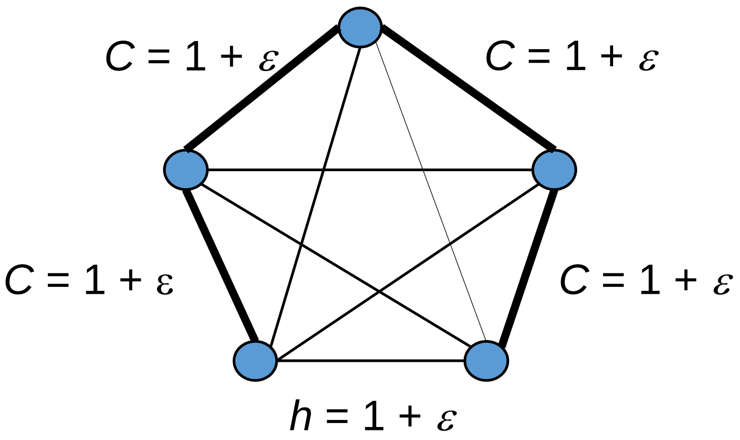
In general, when we have n x m nodes (n in the horizontal direction, and m in vertical), the formula reads

$$\binom{n+m-2}{m-1} = \binom{n+m-2}{n-1}$$



5 by 5 Manhattan network:
70 shortest-hop paths between two opposite corners

routing paths - difficulties



all 10 demands but one with h = 1

all 10 links but four with capacity 1

how should we know that the thick path must be used to get the optimal solution?

DAP (demand allocation problem) – node-link (N-L) notation

Book: problem 4.1.5

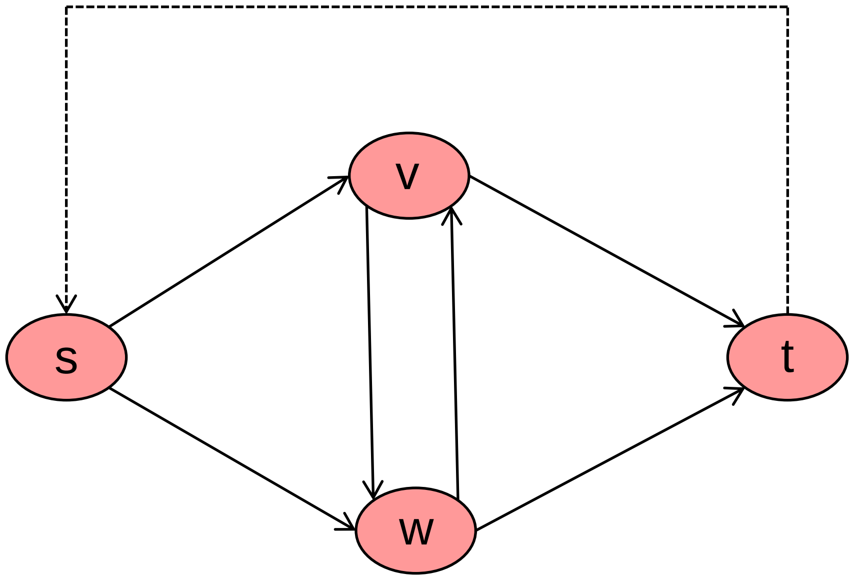
- for directed graphs (link = directed arc)
- arcs: $a = 1,2,\dots,A \quad (a \in \mathcal{A}) \quad C_a$ – capacity of arc a
- nodes: $v = 1,2,\dots,V \quad (v \in \mathcal{V})$
- demands: $d = 1,2,\dots,D \quad (d \in \mathcal{D}) \quad (o(d), t(d))$ – source and destination (termination)
- $\delta^+(v)$ – set of arcs outgoing from node v, $\delta^-(v)$ – set of arcs incoming to node v
- variables
  - x_{ad} continuous flow realizing demand d on arc a

DAP – node-link (N-L) formulation

- objective   minimize z
- constraints
  - $\sum_{a\in\delta^+(v)} x_{ad} = \sum_{a\in\delta^-(v)} x_{ad}$ $v \in \mathcal{V}\setminus\{o(d), t(d)\}, d \in \mathcal{D}$ (1st Kirchoff 's law)
  - $\sum_{a\in\delta^+(o(d))} x_{ad} = \sum_{a\in\delta^-(o(d))} x_{ad} + h_d$ $d \in \mathcal{D}$
  - $\sum_{a\in\delta^+(t(d))} x_{ad} + h_d = \sum_{a\in\delta^-(t(d))} x_{ad}$ $d \in \mathcal{D}$
  - $\sum_{d\in\mathcal{D}} x_{ad} \le C_a + z$ $a \in \mathcal{A}$
  - flow variables x are non-negative integers, z - continuous

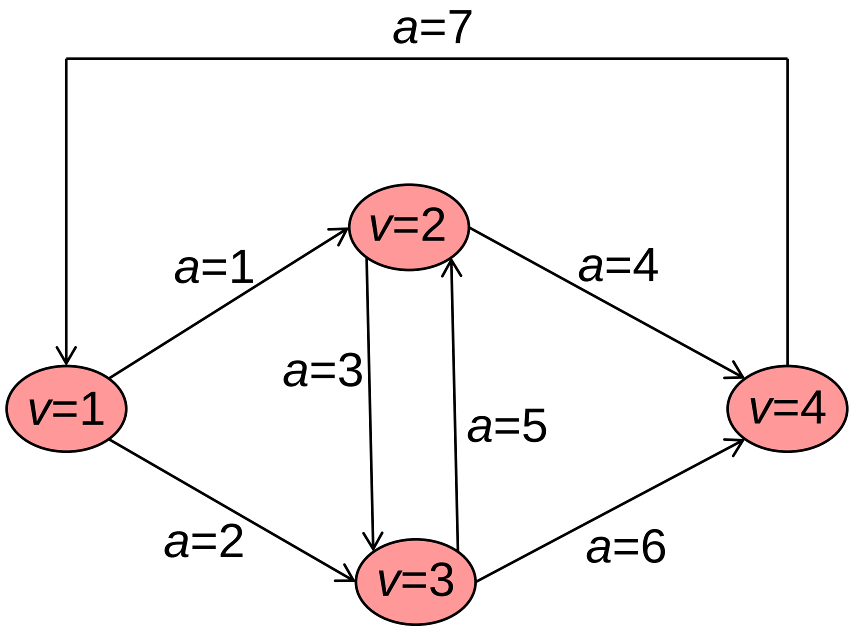
- linear relaxation - simplex algorithm
  - the solution will not in general be integral (we already know this for the undirected case)

N-L formulation - example



$h_{st} = 2$
$C_a = 1$ for all arcs

N-L formulation – example formulation



$d = 1$, $o(1) = 1$, $t(1) = 4$ $h_1 = 2$

$C_a = 1$ for all arcs

minimize z

$$
\begin{array}{lllllllll}
x_{11} + x_{21} & & & & & - x_{71} = & 2 & (v=1) \\
- x_{11} & + x_{31} + x_{41} & - x_{51} & & & = & 0 & (v=2) \\
- x_{21} - x_{31} & & + x_{51} + x_{61} & & & = & 0 & (v=3) \\
 & - x_{41} & & - x_{61} & + x_{71} & = & -2 & (v=4)
\end{array}
$$

$x_{a1} \leq C_a + z$, $a=1,2,\ldots,7$

And in the L-P notation?

N-L formulation - comments

- N–L formulation takes (implicitly) all paths into account
- L–P $\Rightarrow$ N–L : $x_{ad} = \sum_p \delta_{adp} x_{dp}$ (for arc a and demand d)
  - note that L–P works for directed graphs as well
- N–L $\Rightarrow$ L–P : when arc–flows x_{ad} are determined, path–flows x_{dp} can be computed
  - shortest path algorithm applied iteratively to $C_a = x_{ad}$ $(a = 1,2,\dots,A)$ separately for each d
- evolutionary algorithm for integer flows ?