Design of Resilient Multi-layer Core Networks

Multi-commodity Flow Networks Approach to Modelling and Design

Michał Pióro

Institute of Telecommunications
Warsaw University of Technology (Poland)
and
Department of Communication Systems
Lund University (Sweden)

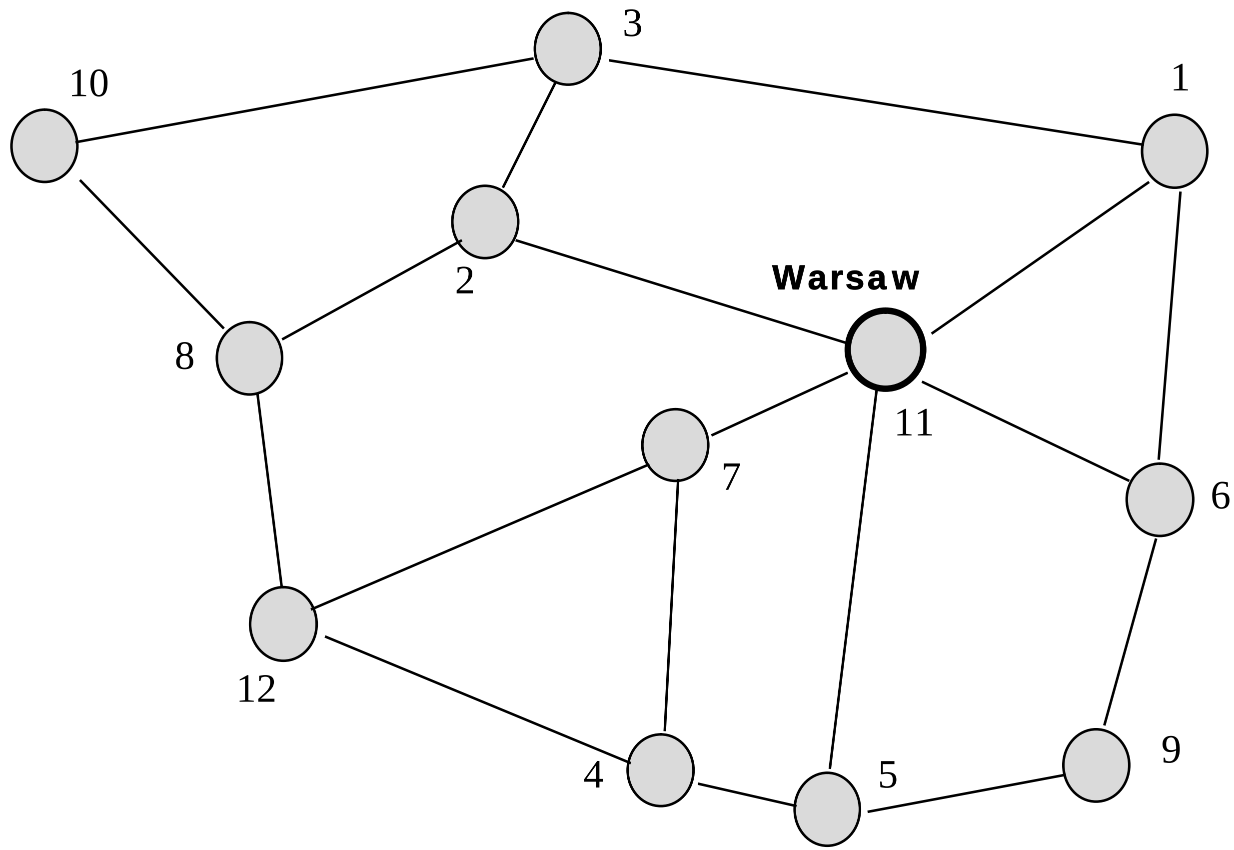
Purpose

- To present basic approachs in network modeling and optimization
  - multi-commodity flow networks
  - routing/reslience/multi-layering

Network – the main object

Telecommunication technologies:

- **IP/OSPF**
- **MPLS**
- **IDN**
- **ATM**
- **SDH**
- **DWDM**



Networks can be really large (50-100 nodes)

Based on:

Michał Pióro and Deepankar Medhi

Routing, Flow, and Capacity Design in Communication and Computer Networks

Morgan Kaufmann Publishers (Elsevier), 2004
770 pages, ISBN 0125571895

- **www.mkp.com**

Auxiliary literature

- R.K.Ahuja et al.: Network Flows: Theory, Algorithms and Applications, Prentice Hall, 1993
- D.P.Bertsekas: Network Optimization: Continuous and Discrete Models, Athena Scientific, 1998
- W.K.Chen: Theory of Nets Flows in Networks, J.Wiley,1990
- R.Bhandari: Survivable Networks Algorithms for Diverse Routing, Kluwer, 1999.
- L.Wosley: Integer Programming, J.Wiley, 1998
- M.Minoux: Mathematical Programming: Theory and Algorithms, J.Wiley, 1986.

- and tons of papers ...

CONTENTS

1. Basic formulations. Dimensioning vs. allocation problems. Link-path vs. node-link formulation. Non-bifurcated flows and other routing restrictions. Modular link capacities and topological design. Importance of LP/MIP/IP models.

2. Shortest path routing (OSPF). MIP formulations. Direct and two-phase approach.

3. Basic resilience mechanisms and related design problems (path diversity, hot-standby, path protection, link protection). LP and MIP formulations.

4. Multi-layer modeling. One-state design problems. Design problems involving reconfiguration on different resource layers.

Dimensioning problem - Link-Path Formulation

- **indices**
  - e=1,2,…,E links
  - d=1,2,…,D demands
  - p=1,2,…,P_d paths for flows realizing demand d
    ($P_{dp} \subseteq \{1,2,...,E\}$ – subset of the set of links indices)

- **constants**
  - δ_{edp} = 1 if e belongs to path p realizing demand d; 0, otherwise
    (δ_{edp} = 1 iff $e \in P_{dp}$)
  - h_d volume of demand d
  - ξ_e unit (marginal) cost of link e
  - M module size of the link capacity

Dimensioning problem – MIP

- **variables**
  - x_{dp} — continuous flow realizing demand d on path p
  - y_e — integer capacity of link e
- **objective** — minimize $\mathbf{F}(\mathbf{y}) = \Sigma_e \xi_e y_e$
- **constraints**
  - $\Sigma_p x_{dp} = h_d$ — $d=1,2,\dots,D$
  - $\Sigma_d \Sigma_p \delta_{edp} x_{dp} \leq M y_e$ — $e=1,2,\dots,E$
  - all variables are *non-negative*

How to solve this NP-hard problem?

Simple dimensioning problem – LP relaxation

- **variables**
  - x_{dp} flow realizing demand d on path p
  - y_e capacity of link e

- **objective** minimize $\mathbf{F}(\mathbf{y}) = \Sigma_e \xi_e y_e$

- **constraints**
  - $\Sigma_p x_{dp} = h_d$ d=1,2,…,D
  - $\Sigma_d \Sigma_p \delta_{edp} x_{dp} \leq y_e$ e=1,2,…,E (=)
  - all variables are *continuous and non-negative*

How to solve this problem?

LP relaxation - solution

- **constraints**
  - $\Sigma_p\ x_{dp} = h_d$ d=1,2,…,D
  - $\Sigma_d\ \Sigma_p\ \delta_{edp}x_{dp} = y_e$ e=1,2,…,E
- **F** $= \Sigma_e\ \xi_e\ \Sigma_d\ \Sigma_p\ \delta_{edp}x_{dp} = \Sigma_d\ \Sigma_p\ (\Sigma_e\ \xi_e\ \delta_{edp})x_{dp} = \Sigma_d\ \Sigma_p\ \kappa_{dp}x_{dp} = \Sigma_d\ \zeta_d h_d$

  where

  κ_{dp} - cost of path p of demand d

  ζ_d – cost of the cheapest (shortest with respect to ξ_e) path of demand d
- **solution**: put the whole demand on the shortest path(s)

(Capacitated) flow allocation problem

- **indices**
  - $d=1,2,\ldots,D$ demands
  - $p=1,2,\ldots,P_d$ paths for flows realizing demand d
  - $e=1,2,\ldots,E$ links

- **constants**
  - h_d volume of demand d
  - c_e capacity of link e
  - δ_{edp} = 1 if e belongs to path p realizing demand d; 0, otherwise

Capacitated flow allocation problem – Link-Path formulation

- **variables**
  - x_{dp} flow realizing demand d on path p
- **constraints**
  - $\Sigma_p \; x_{dp} = h_d$ $\quad d=1,2,\ldots,D$
  - $\Sigma_d \, \Sigma_p \, \delta_{edp} x_{dp} \leq c_e$ $\quad e=1,2,\ldots,E$
  - flow variables are *continuous and non-negative*

D+E property:

If the problem is feasible then there exists a solution with at most D+E non-zero flows.

so far we have been using <u>link-path formulation</u>
applicable for both undirected and directed graphs

indices

- d=1,2,…,D demands
- v=1,2,... ,V nodes
- e=1,2,...,E links (directed arcs)

- **constants**
  - h_d volume of demand d
  - s_d, t_d source, sink node of demand d
  - a_{ev} = 1 if link e originates at node v; 0, otherwise
  - b_{ev} = 1 if link e terminates in node v; 0, otherwise
  - c_e capacity of link e

for directed graphs (applicable to undirected graphs after some transformation)

Node-link formulation

- **variables**
  - $x_{ed} \geq 0$ flow of demand d on link e

- **constraints**
  - $$\Sigma_e a_{ev} x_{ed} - \Sigma_e b_{ev} x_{ed} = \begin{cases} h_d & \text{if } v = s_d \\ 0 & \text{if } v \neq s_d, t_d \\ -h_d & \text{if } v = t_d \end{cases}$$
    $v=1,2,...,V \quad d=1,2,\ldots,D$
  - $\Sigma_d x_{ed} \leq c_e$ $e=1,2,\ldots,E$

Aggregated node-link formulation

- **indices**
  - v,w=1,2,... ,V   nodes
  - e=1,2,...,E   arcs
- **constants**
  - h_{vw} volume of demand d originating at node v and terminating at node w
  - H_v $= \Sigma_{w \neq v} h_{vw}$ total demand volume originating at node v
  - a_{ev} = 1 if link e originates at node v; 0, otherwise
  - b_{ev} = 1 if link e terminates in node v; 0, otherwise
  - c_e capacity of link e

Aggregated node-link formulation

- **variables**
  - $x_{ev} \geq 0$ flow realizing all demands originating at node v on link e
- **constraints**
  - $\Sigma_e a_{ev} x_{ev} = H_v$ v=1,2,...,V (redundant)
  - $\Sigma_e a_{ew} x_{ev} - \Sigma_e b_{ew} x_{ev} = - h_{vw}$ v,w=1,2,...,V $v \neq w$
  - $\Sigma_v x_{ev} \leq c_e$ e=1,2,…,E

Number of variables and constraints

	#variables	#constraints
L-P	$P \times V(V-1) = O(V^2)$	$V(V-1) + (k \times V)/2 = O(V^2)$
N-L	$(k \times V \times V(V-1))/2 = O(V^3)$	$V \times V(V-1) + (k \times V)/2 = O(V^3)$
A/N-L	$(k \times V \times V)/2 = O(V^2)$	$V \times V + (k \times V)/2 = O(V^2)$

Remarks

1. Still N-L can be faster than A/N-L.
2. There must be a way for path generation in L-P formulations.

Path generation

- note that in the **link-path formulation** the lists of paths are predefined
- using of full lists is not realistic (exponential number of paths)
- fortunately, path generation method can be applied (general method called „column generation" in LP, related to revised Simplex)
- optimal dual multipliers π_e assocoated with capacity constraints are used to generate new shortest paths
- the paths can be generated using Dijkstra, or some other shortest path algorithm, e.g., with limited number of hops
- **link-path formulation with PG** can be superior to using the **node-link formulation**:
  - less variables and constraints
  - a limited number of non-zero flows is used in a Simplex (basic) solution (e.g., at most D+E for the allocation problem)
  - in the N-L formulation we do not control the paths – have to use them all; in the L-P formulation we can control the paths

Path generation: Flow allocation problem – LP formulation

- **variables**
  - x_{dp} flow realizing demand d on path p
- **constraints**
  - $\Sigma_p x_{dp} = h_d$ $d=1,2,\dots,D$
  - $\Sigma_d \Sigma_p \delta_{edp} x_{dp} \leq c_e$ $e=1,2,\dots,E$
  - flow variables are continuous and non-negative

- D+E non-zero flows at most
  - depending on the number of saturated links
  - if all links unsaturated: D flows only!

- How to define the proper lists of paths? **By path generation.**

Capacitated flow allocation problem – adjusted formulation

- **variables**
  - x_{dp} flow realizing demand d on path p
  - z auxiliary variable

- **objective** minimize z

- **constraints**
  - $\Sigma_p x_{dp} = h_d$ $d=1,2,\dots,D$ (λ_d - unconstrained)
  - $\Sigma_d \Sigma_p \delta_{edp} x_{dp} \le c_e + z$ $e=1,2,\dots,E$ ($\pi_e \ge 0$)
  - flow variables are continuous and non-negative, z is continuous

Dual

- $L(\mathbf{x},z;\pi,\lambda) = z + \Sigma_d \lambda_d(h_d - \Sigma_p x_{dp}) + \Sigma_e \pi_e(\Sigma_d \Sigma_p \delta_{edp}x_{dp} - c_e - z)$
  - $x_{dp} \geq 0$ for all (d,p)

- $W(\pi,\lambda) = \min_{\mathbf{x}\geq\mathbf{0},z} L(\mathbf{x},z;\pi,\lambda)$

- **Dual**
  - maximize $W(\pi,\lambda) = \Sigma_d \lambda_d h_d - \Sigma_e \pi_e c_e$
  - subject to
    - $\Sigma_e \pi_e = 1$
    - $\lambda_d \leq \Sigma_e \delta_{edp}\pi_e$ d=1,2,…,D p=1,2,...,P_d
    - $\pi_e \geq 0$ e=1,2,...,E

Path generation - the reason

- **Dual**
  - maximize $\Sigma_d \lambda_d h_d - \Sigma_e \pi_e c_e$
  - subject to
    - $\Sigma_e \pi_e = 1$
    - $\lambda_d \leq \Sigma_e \delta_{edp} \pi_e$ $d=1,2,\ldots,D$ $p=1,2,\ldots,P_d$
    - $\pi_e \geq 0$ $e=1,2,\ldots,E$

- **if we can find a path shorter than $S_e\, d_{edp} p_e$* then we will get a more constrained dual problem and hence have a chance to improve (decrease) the optimal dual value**

- shortest path algorithm can be used for finding shortest paths with respect to p*

Path generation - how it works

- We can start with only one single path on the list for each demand (P_d = 1 for all d).
- Then we solve the extended problem and add one shortest path for each demand d (if such a path, i.e., a path shorter than all teh paths on the current list for demand d, exists).
- This process will terminate typically after only several steps.
- Cycling may occur, so it is better not to remove paths.

Section 5.4.2

Path diversity (works for disjoint paths)

- **variables**
  - x_{dp} — flow realizing demand d on path p
  - y_e — capacity of link e

- **constraints**
  - $\Sigma_p x_{dp} = h_d$ — $d=1,2,\dots,D$
  - $x_{dp} \leq h_d / n_d$ — $d=1,2,\dots,D$, $p=1,2,\dots,P_d$
  - $\Sigma_d \Sigma_p \delta_{edp} x_{dp} = y_e$ — $e=1,2,\dots,E$
  - all variables non-negative

In node-link formulation?

Generalized path diversity

variables

- x_{dp} — flow realizing demand d on path p
- y_e — capacity of link e

- **constraints**
  - $\Sigma_p \; x_{dp} = h_d$ — $d=1,2,\ldots,D$
  - $\Sigma_p \; \delta_{edp} x_{dp} \le h_d / n_d$ — $e=1,2,\ldots,E$, $d=1,2,\ldots,D$
  - $\Sigma_d \Sigma_p \; \delta_{edp} x_{dp} = y_e$ — $e=1,2,\ldots,E$
  - all variables non-negative

In node-link formulation:

$x_{ed} \le h_d / n_d$ — $e=1,2,\ldots,E$, $d=1,2,\ldots,D$

Single path allocation (non-bifurcated flows)

NP-hard

- **variables**
  - u_{dp} binary flow variable corresponding to demand d and path p

- **constraints**
  - $\Sigma_p u_{dp} = 1$ d=1,2,…,D
  - $\Sigma_d \Sigma_p \delta_{edp}(h_d u_{dp}) \leq c_e$ e=1,2,…,E
  - **u** binary

In node-link formulation?

Single-path allocation: N-L formulation

- **variables**
  - $u_{ed} \geq 0$ binary variable associated with flow of demand d on link e

- **constraints**
  - $\Sigma_e a_{ev} u_{ed} - \Sigma_e b_{ev} u_{ed}$
    - = 1 if $v = s_d$
    - = 0 if $v \neq s_d, t_d$
    - = -1 if $v = t_d$

    $v=1,2,...,V \quad d=1,2,…,D$
  - $\Sigma_d h_d u_{ed} \leq c_e$ $\quad e=1,2,…,E$

Now, hop limit can be introduced (how?). Very easy in the L-P formulation.

Other extensions

- There are many other extensions (Chapter 4)
  - lower-bounded non-zero flows
  - equal split to at most k paths
  - limited number of hops.
- Not all of them can be expressed in node-link formulation.
- Think what is the essential diffrence between N-L and L-P formulations. (Hint: only paths with certain properties can be used, e.g., limited number of hops.)

Dimensioning problem – MIP variants (NP-hard)

- **Modular capacities**
  - minimize $\mathbf{F}(\mathbf{y}) = \sum_e \xi_e y_e$
  - $\sum_d \sum_p \delta_{edp} x_{dp} \leq M \cdot y_e$ e=1,2,…,E y_e integer - MIP
  - $\sum_d \sum_p \delta_{edp} x_{dp} \leq M \cdot y_e$ e=1,2,…,E x_{dp}, y_e integer – IP

- **Multiple modules**
  - minimize $\mathbf{F}(\mathbf{y}) = \sum_e \sum_k \xi_{ek} y_{ek}$
  - $\sum_d \sum_p \delta_{edp} x_{dp} \leq \sum_k M_k \cdot y_{ek}$ e=1,2,…,E y_{ek} integer – MIP

- **Incremental modular capacities**
  - minimize $\mathbf{F}(\mathbf{y}) = \sum_e \xi_{ek} u_{ek}$
  - $\sum_d \sum_p \delta_{edp} x_{dp} \leq \sum_k m_k \cdot u_{ek}$ e=1,2,…,E u_{ek} binary – MIP
  - $u_{e1} \geq u_{e2} \geq \ldots \geq u_{eK}$ e=1,2,…,E

Dimensioning problem – convex and concave variants

- **objective** minimize $\mathbf{F}(\mathbf{y}) = \Sigma_e \xi_e f(y_e)$
  - $y_e = \Sigma_d \Sigma_p \delta_{edp} x_{dp}$
  - f(z) – convex (penalty, delay) - CXP
  - f(z) – concave (dimensioning function) – CVP

Convex – LP approximation (easy)

Concave – MIP approximation (difficult)

Benders' decomposition: Dimensioning problem (LP or MIP)

- **variables**
  - x_{dp} flow realizing demand d on path p (continuous)
  - y_e link capacities, continuous (LP) or integer (MIP)

- **objective** minimize $\Sigma_e \xi_e y_e$

- **constraints**
  - $\Sigma_p x_{dp} = h_d$ d=1,2,…,D
  - $\Sigma_d \Sigma_p \delta_{edp} x_{dp} \leq y_e$ e=1,2,…,E

Benders' decomposition

- iterative process
  - eliminate flow variables **x**
  - introduce new constraints on **y**

Section 5.4.3

BD scheme

- **Step 0**
  - W := { $y_e \geq 0$: e=1,2,...,E } (set of inequality constraints on **y**)
- **Step 1: solve master problem (y*)**
  - minimize $\Sigma_e \xi_e y_e$
  - subject to W
- **Step 2: perform the feasibility test (p*, l*)**
  - maximize W(p, l) = $\Sigma_d \lambda_d h_d - \Sigma_e \pi_e y_e^*$
  - subject to
    - $\Sigma_e \pi_e = 1$
    - $\lambda_d \leq \Sigma_e \delta_{edp} \pi_e$ d=1,2,…,D p=1,2,...,P_d
    - $\pi_e \geq 0$ e=1,2,...,E

  if W(p*, l*) > 0 then add inequality $\Sigma_d \lambda_d^* h_d - \Sigma_e \pi_e^* y_e \leq 0$ to W
- **Step 3: if new inequality is added then go to Step 1, otherwise STOP**

BD scheme - remarks

- **In MP there are no flow variables**
- **FT is a dual to the flow allocation problem:**
  - **objective** minimize z
  - **constraints**
    - $\Sigma_p\, x_{dp} = h_d$ $d=1,2,\ldots,D$ (λ_d - unconstrained)
    - $\Sigma_d\, \Sigma_p\, \delta_{edp} x_{dp} \leq y_e^{*} + z$ $e=1,2,\ldots,E$ ($\pi_e \geq 0$)
    - flow variables are continuous and non-negative, z is continuous
- **Can be much effective than the direct approach for the MIP version**

Topological design – fixed charge model

NP-hard

- **variables**
  - x_{dp} flow of demand d on path p
  - y_e capacity of link e
  - u_e =1 if link e is installed; 0, otherwise

- **objective** minimize $\mathbf{F} = \Sigma_e \xi_e y_e + \Sigma_e \kappa_e u_e$

- **constraints**
  - $\Sigma_p x_{dp} = h_d$ d=1,2,…,D
  - $\Sigma_d \Sigma_p \delta_{edp} x_{dp} = y_e$ e=1,2,…,E
  - $y_e \leq M_e u_e$ e=1,2,…,E
  - **y** and **x** non-negative, and **u** binary

Why LPs, MIPs, and IPs are so important?

- in practice only LP guarantees efficient solutions
- decomposition methods are available for LPs
- MIPs and IPs can be solved by general solvers by the branch-and-cut method, based on LP
  - CPLEX, XPRESS
  - sometimes very efficiently
- otherwise, we have to use (frequently) unreliable stochastic meta-heuristics (sometimes specialized heuristics)

Shortest Path Routing (IP/OSPF)

- Links are assigned admistrative weights (simplest $w_e = 1$)
  - w_e weight (metric) of link e, $\mathbf{w} = (w_1,w_2,...,w_E)$.
- Suppose that for each demand d there is only one shortest path with respect to **w**.
- Then, this path is used to carry the whole demand volume h_d.

- **indices**
  - $d=1,2,\ldots,D$ demands
  - $p=1,2,\ldots,P_d$ paths for flows realizing demand d
  - $e=1,2,\ldots,E$ links
- **constants**
  - h_d volume of demand d
  - c_e capacity of link e
  - δ_{edp} = 1 if e belongs to path p realizing demand d; 0, otherwise

Chapter 7

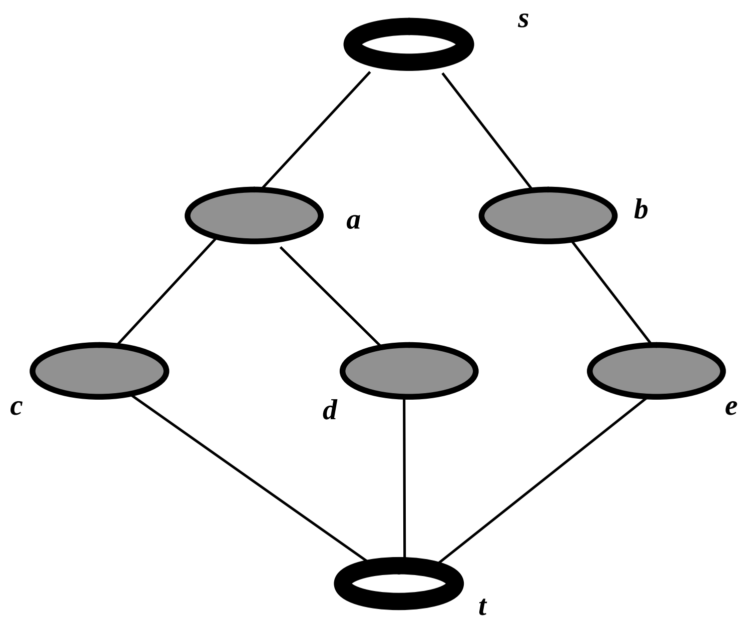
Shortest Path Routing

NP-hard

- **variables**
  - w_e weight (metric) of link e, $\mathbf{w} = (w_1,w_2,...,w_E)$
  - $x_{dp}(\mathbf{w})$ ECMP flow induced by metric system $\mathbf{w}$ on path (d,p)
- **constraints**
  - $\Sigma_p\ x_{dp}(\mathbf{w}) = h_d$ d=1,2,…,D
  - $\Sigma_d\,\Sigma_p\,\delta_{edp}x_{dp}(\mathbf{w}) \leq c_e$ e=1,2,…,E (*)
  - $\mathbf{w} \in W$

Penalty function can be used instead of (*).

ECMP (Equal Cost Multi-Path) rule



ECMP (Equal-split rule)

- ECMP flow
  - flow to a destination outgoing from a node is equally split into the outgoing links belonging to the shortest paths to this destination

- all link weights w_e equal to 1
- ECMP flow
  - from s to t
  - from t to s

How to compute ECMP flows for given weights?

- **indices**
  - $d=1,2,\ldots,D$ demands
  - $p=1,2,...,P_d$ paths of demand d
  - $e=1,2,\ldots,E$ links
  - $s,t,v=1,2,...,V$ nodes

- **constants**
  - h_{vt} demand volume from node v to node t
  - w_e link weights
  - $i(d)$, $j(d)$ end nodes of d
  - $i(e)$, $j(e)$ starting and terminating node of link e

can be done algorithmically

(Algorithm 7.1, Section 7.3.3)

ECMP flows for given weights

- **variables**
  - r_d lenght of the shortest path of demand d
  - z_{et} = 0 if link e is on a shortest path to node t (1, otherwise)
  - x_{et} flow to node t on link e
  - y_{vt} common value of non-zero flow from node v to node t assigned to links outgoing from v

ECMP flows for given weights - LP

- **maximize** $E\,\Sigma_d\, r_d + \Sigma_e \Sigma_t\, z_{et}$
- **constraints**
  - $r_d \leq \Sigma_e\, \delta_{edp} w_e$ — $d=1,2,\ldots,E,\ p=1,2,\ldots,P_d$
  - $z_{et} \leq \Sigma_e\, \delta_{edp} w_e - r_d$ — $d=1,2,\ldots,E,\ p=1,2,\ldots,P_d,\ t=j(d),\ e \in (d,p)$
  - $0 \leq z_{et} \leq 1$ — $e=1,2,\ldots,E,\ t=1,2,\ldots,V$
  - $\Sigma_{\{e:\ j(e)=t\}}\, x_{et} = \Sigma_{s\neq t}\, h_{st}$ — $t=1,2,\ldots,V$
  - $\Sigma_{\{e:\ i(e)=v\}}\, x_{et} - \Sigma_{\{e:\ j(e)=v\}}\, x_{et} = h_{vt}$ — $t,v=1,2,\ldots,V,\ t\neq v$
  - $0 \leq y_{i(e)t} - x_{et} \leq z_{et}\Sigma_v\, h_{vt}$ — $t=1,2,\ldots,V,\ e=1,2,\ldots,E$
  - $x_{et} \leq (1-z_{et})(\Sigma_v\, h_{vt})$ — $t=1,2,\ldots,V,\ e=1,2,\ldots,E$

we assume that if two paths have different length then they differ by at least one

ECMP flows for given weight - N-L formulation

- **maximize** $V\Sigma_s \Sigma_t r_{st} + \Sigma_e \Sigma_t z_{et}$
- **constraints**
  - $r_{st} \leq w_e + r_{j(e)t}$ s=1,2,…,V, t=1,2,...,V, i(e) = s
  - $z_{et} \leq w_e + r_{j(e)t} - r_{i(e)t}$ s=1,2,…,V, t=1,2,...,V
  - $0 \leq z_{et} \leq 1$ e=1,2,…,E, t=1,2,...,V
  - $\Sigma_{\{e:\, j(e)=t\}} x_{et} = \Sigma_{s\neq t} h_{st}$ t=1,2,...,V
  - $\Sigma_{\{e:\, i(e)=v\}} x_{et} - \Sigma_{\{e:\, j(e)=v\}} x_{et} = h_{vt}$
    t,v=1,2,…,V, t≠v
  - $0 \leq y_{i(e)t} - x_{et} \leq z_{et}\Sigma_v h_{vt}$ t=1,2,…,V, e=1,2,…,E
  - $x_{et} \leq (1-z_{et})(\Sigma_v h_{vt})$ t=1,2,…,V, e=1,2,…,E

Shortest Path Routing – MIP aggregated node-link formulation

- **indices**
  - t,v,s=1,2,…,V nodes
  - e=1,2,…,E links

- **constants**
  - h_{vt} demand from node v to node t
  - i(e), j(e) starting and terminating node of link e
  - M large number
  - c_e capacity of link e

Shortest Path Routing – MIP formulation

- **variables**
  - w_e weight (metric) of link e, $\mathbf{w} = (w_1,w_2,...,w_E)$
  - r_{vt} length of the shortest path from v to t ($r_{vv} = 0$)
  - x_{et} flow to node t on link e
  - y_{vt} common value of non-zero flow from node v to node t assigned to links outgoing from v
  - u_{et} binary variable equal to 1 iff link e is on the shortest path to node t ($u_{et} = 1 - z_{et}$)

- **no objective**

  (can be added – e.g., minimization of maximal link utilization)

- **constraints**
  - $\Sigma_{\{e:\ j(e)=t\}}\, x_{et} = \Sigma_{s\neq t}\, h_{st}$ t=1,2,…,V
  - $\Sigma_{\{e:\ i(e)=v\}}\, x_{et} - \Sigma_{\{e:\ j(e)=v\}}\, x_{et} = h_{vt}$ t,v=1,2,…,V, t≠v
  - $\Sigma_t\, x_{et} \leq c_e$ e=1,2,…,E
  - $0 \leq y_{i(e)t} - x_{et} \leq (1 - u_{et})\Sigma_v\, h_{vt}$ t=1,2,…,V, e=1,2,…,E
  - $x_{et} \leq u_{et}\, \Sigma_v\, h_{vt}$ t=1,2,…,V, e=1,2,…,E
  - $0 \leq r_{j(e)t} + w_e - r_{i(e)t} \leq (1 - u_{et})M$ t=1,2,…,V, e=1,2,…,E
  - $1 - u_{et} \leq r_{j(e)t} + w_e - r_{i(e)t}$ t=1,2,…,V, e=1,2,…,E
  - $w_e \geq 1$ e=1,2,…,E
  - $\mathbf{w} \in W$

MIP formulation - extensions

- **limited split**
  - $\Sigma_{\{e:\ i(e)=t\}} u_{et} \le n$ $\quad$ $t,v=1,2,\dots,V,\ t \ne v$
- **auxiliary objective function**
  - minimize $\quad t$
  - subject to $\quad t \ge (\Sigma_t x_{et}) / c_e \quad e=1,2,\dots,E$
- **dimensioning problem**
  - minimize $\mathbf{F} = \Sigma_e \xi_e y_e$
  - use variables y_e instead of c_e
- **installation costs κ_e – as well**

Approaches

- Dual approach
- Two-phase approach
- Iterative approach
- Direct MIP formulation + branch and cut (Holmberg and Yuan, Ben Ameur and E. Gourdin, Tomaszewski and Pioro)
- Heuristics (SAN)

Direct approachs

- Branch-and-cut with the direct MIP formulation
- SAN (and other meta-heuristics)
- Dual approach
- Other B&C approaches (Holmberg and Yuan, Ben Ameur and E. Gourdin, Bley)

ECMP Optimisation Problem

solution space: **W**

evaluation function: maximal overload $\mathbf{F}(\mathbf{w}) = \max \{\underline{y}_e(\mathbf{w})/c_e : e=1,2,...,E\}$

find $\mathbf{w} \in \mathbf{W}$ with minimizing $\mathbf{F}(\mathbf{w})$

$N(\mathbf{w}) \subseteq \mathbf{W}$ – change one weight by ±1 (or some more drastic change)

One must compute the ECMP flows.

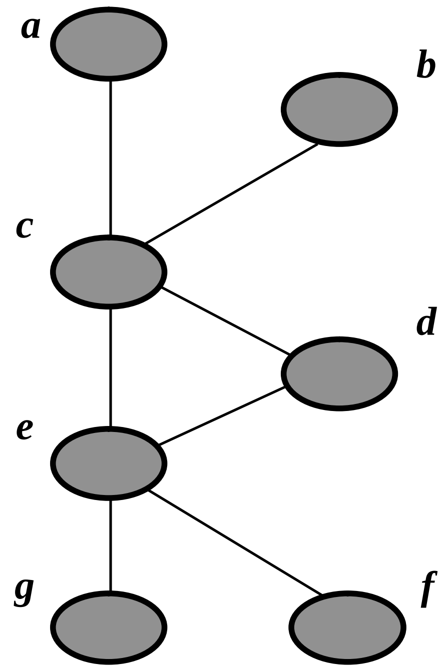
Two-phase approach

- **Phase 1:**

  Solve capacitated single-path allocation problem (MIP).

- **Phase 2:**

  Find (continuous, ≥ 1) weights generating the paths used in the solution of Phase 1 (easy to find using an LP when exist) and convert them to integers.



Infeasible paths

Dual approach

- Skip weights and find single path routing:

variables

x_{dp} continuous flow on path (d,p)

maximize F $= \Sigma_e (c_e - \Sigma_d \Sigma_p \delta_{edp} x_{dp})$

constraints

$\Sigma_p u_{dp} = 1$ d=1,2,…,D

$\Sigma_d \Sigma_p \delta_{edp} x_{dp} \le c_e$ e=1,2,…,E (π_e)

- Solve the dual problem and use weights $w_e = 1 + \pi_e$.

Dual approach – dual problem

- **variables**
  - λ_d unrestricted in sign
  - π_e non-negative

- **objective**

  maximize $\mathbf{G} = \Sigma_d \lambda_d h_d - \Sigma_e (1 + \pi_e)c_e$

- **constraints**
  - $\lambda_d \leq \Sigma_e \delta_{edp}(1 + \pi_e)$ d=1,2,…,E, p=1,2,…,P_d

When optimal p determine single paths for all demands we are done!

- $w_e = 1 + \pi_e$ are the optimal weights (can be easily made integer)

Two-phase approach for single shortest paths

- **Phase 1:**

  Solve capacitated single-path allocation problem (MIP).

- **Phase 2:**

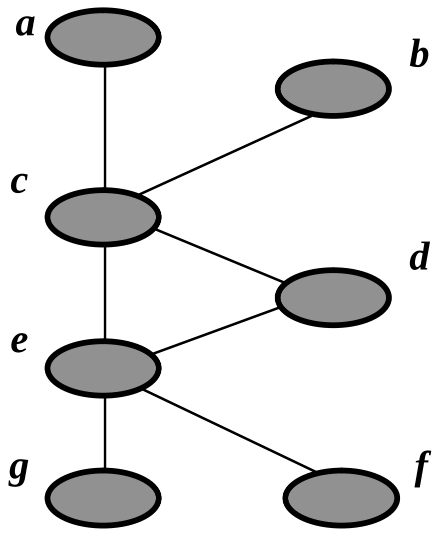
  Find (continuous, ≥ 1) weights generating the paths used in the solution of Phase 1 (easy to find using an LP when exist) and convert them to integers.

Phase 1 – single path routing

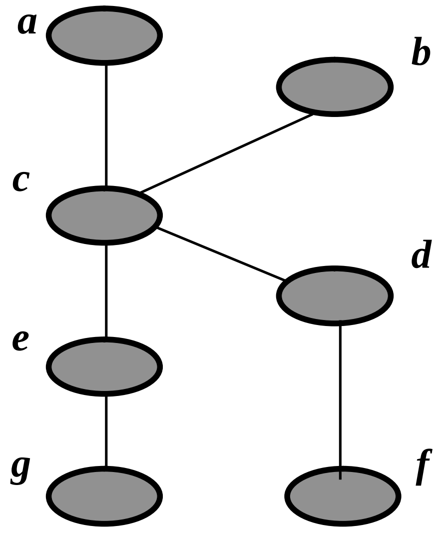
- **variables**
  - u_{dp} binary flow variable for demand d and path p
- **constraints**
  - $\Sigma_p u_{dp} = 1$ d=1,2,…,D
  - $\Sigma_d h_d \Sigma_p \delta_{edp} u_{dp} \leq c_e$ e=1,2,…,E

Phase 1 – single path routing

- It can happen that Phase 1 results in infeasible paths.
- The chance for feasible paths can be increased by finding a solution fulfilling the necessary condition (see Section 7.4.2 and the use of SAL).



Infeasible paths



Feasible paths

Phase 2 – inverse shortest path problem

- **indices**
  - d=1,2,…,D
  - p=0,1,…,P_d (p=0 path that is supposed to be the unique shortest path)
  - e=1,2,…,E
- **variables**
  - w_e weights
- **constraints**
  - $\Sigma_e \delta_{ed0} w_e + 1 \leq \Sigma_e \delta_{edp} w_e$ d=1,2,…,D, p=1,2,…,P_d
  - $w_e \geq 1$ e=1,2,…,E

the shortest path differs by at least 1 from the other paths

- Simple generation of two shortest paths applied to account for all paths
  - If there are paths that make ths current path system infeasible – add them to the lists and continue
- Weights can be made integer.

Phase 1 – revisited

- Path graph: vertices $G = \{P_{dp}: d=1,2,...,D, p=1,2,...,P_d\}$
  links: between paths which do not fulfill the necessary condition
- Find maximal cliques C in the graph
- If $C \subseteq G$ is such a clique, add a constraint:
  - $\Sigma_{P_{dp} \in C}\, u_{dp} \le 1$ to the problem of Phase 1
- Use heuristics to find maximal cliques (NP-complete itself)

Phase 2 – revisited

- If Phase 2 is infeasible then find the minimal subset C of infeasible rows in the problem dual to the Phase 2 problem (i.e., a *hyperlink* in the path-graph) and add
  - $\Sigma_{Pdp \in C} u_{dp} \leq |C| - 1$ to the problem of Phase 1
  - a function in CPLEX

  and iterate
- It is well known that there may be a set of n paths with any (n-1)-element subset realizeable (e.g., for n=4)
- Some other, much stronger, necessary conditions exist (Holmberg and Broström)

A remark

- Single path allocation problem is NP-complete itself
- In practice, however, much easier to solve by branch-and-cut that the direct MIP

Resilient (robust) design

- Failures of links and nodes are taken into account at the design stage
  - failure scenarios are assumed
  - spare capacity (on top of normal capacity) is provided in a cost effective way
  - demands (flows) are restored when failure occurs

Chapter 9

Resilient design: indices and constants

- **indices**
  - $d=1,2,\ldots,D$ — demands
  - $p=1,2,\ldots,P_d$ — paths for flows realizing demand d
  - $e=1,2,\ldots,E$ — links
  - $s=0,1,\ldots,S$ — failure situations (0 - nominal state)
- **constants**
  - h_{ds} — volume of demand d to be realized in situation s
  - ξ_e — unit cost of link e
  - δ_{edp} — = 1 if e belongs to path p realising demand d; 0, otherwise
  - α_{es} — link failure coefficient of link e in situation s (binary: $\alpha_{es} \in \{0,1\}$, or fractional: $0 \le \alpha_{es} \le 1$)
  - θ_{dps} — $= \prod_{\{e:\, \delta edp=1\}} \alpha_{es} = \min\{ \alpha_{es}: e \in P_{dp} \}$ (for binary α_{es})

Robust network design through path diversity

- **variables**
  - x_{dp0} flow realising demand d on path j in nominal state 0
  - y_e capacity of link e

- **objective** minimize $\mathbf{F}(\mathbf{y}) = \Sigma_e \xi_e y_e$

- **constraints**
  - $\Sigma_p \theta_{dps} x_{dp0} \geq h_{ds}$ d=1,2,…,D , s=0,1,…,S
  - $\Sigma_d \Sigma_p \delta_{edp} x_{dp0} \leq y_e$ e=1,2,…,E
  - all variables non-negative (some integer)

$$\theta_{dps} = \prod_{\{e:\ \delta edp = 1\}} \alpha_{es}$$

Robust network design problem DR-U: unrestricted flow reallocation

- **variables**
  - x_{dps} flow realising demand d on path p in situation s
  - y_e capacity of link e
- **objective** minimize $\mathbf{F}(\mathbf{y}) = \Sigma_e \xi_e y_e$

$\Sigma_d \Sigma_p \delta_{edp} x_{dps} \leq \alpha_{es} M y_e$

- **constraints**
  - $\Sigma_p x_{dps} = h_{ds}$ d=1,2,…,D, s=0,1,…,S
  - $\Sigma_d \Sigma_p \delta_{edp} x_{dps} \leq \alpha_{es} y_e$ e=1,2,…,E, s=0,1,…,S
  - all variables non-negative (some can be integer)

shared protection capacity, reused released capacity

There are two failure situations: s=1,2
in situation s=1 the first path is failed
in situation s=2 the second path is failed
the third path is available in both situations

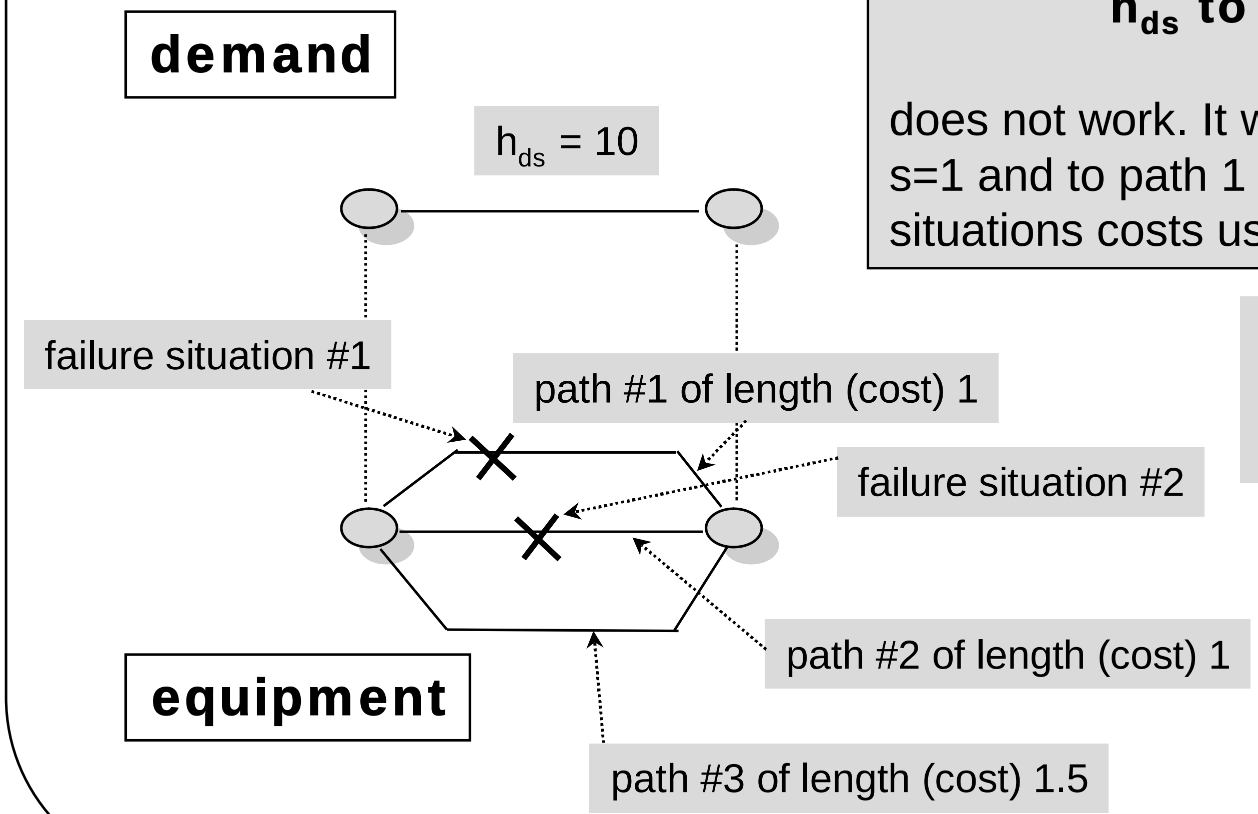
An issue

This example illustrates that the following extension of the LP allocation rule (all variables continuous, identity dim. function):

in situation s allocate all the demanded capacity h_{ds} to its cheapest available path

does not work. It would cost us 20 (allocate h_{ds}=10 to path 2 in s=1 and to path 1 in s=2). Allocating h_{ds}=10 to path 3 in both situations costs us only 15.

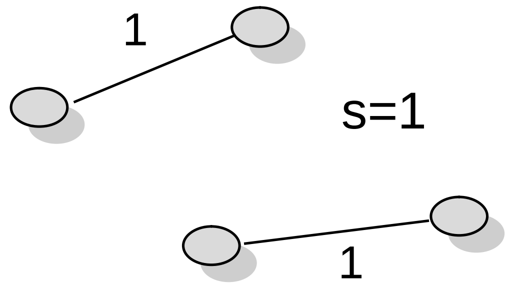
This is because the available capacity of the Layer 1 links is shared among flows in different situations!

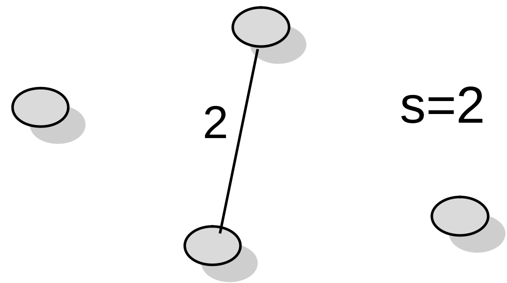


Another issue: bifurcated optimal flows

In general the optimal flows for some demands have to be split among more than one path.

demands

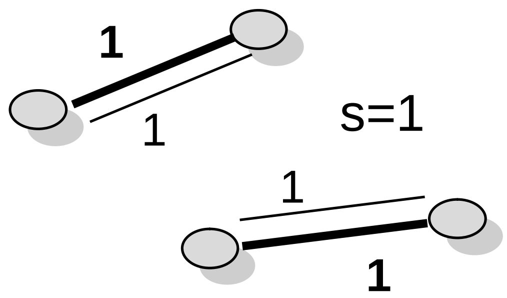


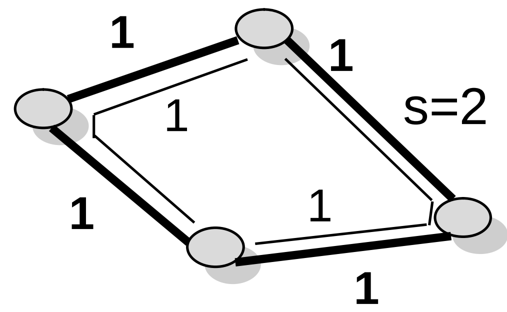


demands

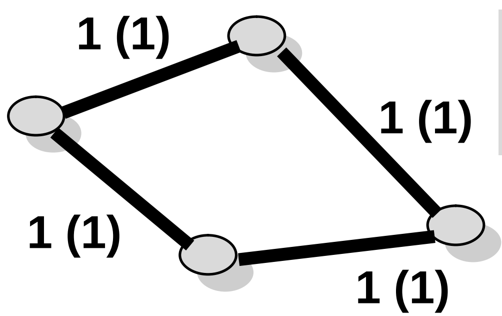
---

equipment





capacities and flows



minimal cost F(y) = 4
(non-bifurcated flows: F(y) = 5)

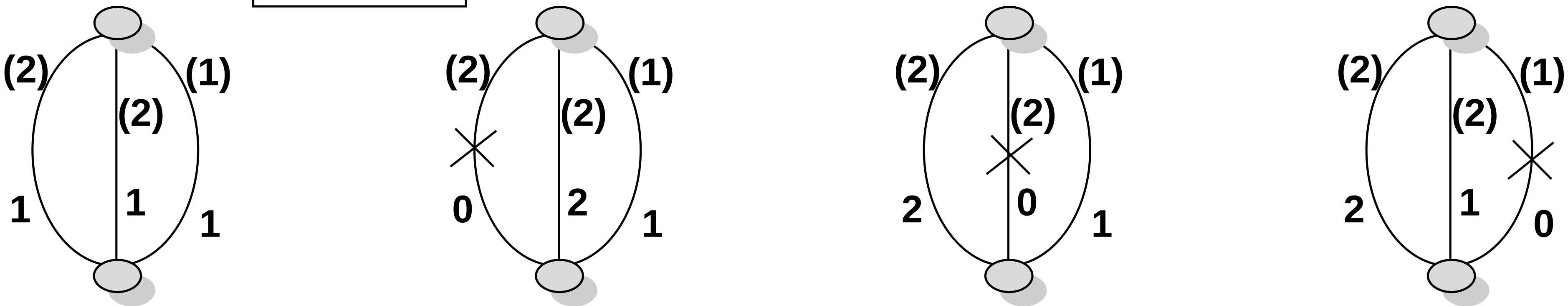
capacities and costs ( )

Another issue - cntd.

Optimal flow is bifurcated: optimal cost of links = 5 (minimal cost with non-bifurcated flow = 6).

link marginal costs are all equal to 1
link capacities: (y_e)
flows: x_{dps}

Layer 1



demand h_{d1}=3 | demand h_{d2}=3 | demand h_{d3}=3 | demand d_{d4}=3

s=1 | s=2 | s=3 | s=4

integer flows; for continuous flows cost = 4.5

Protection/restoration mechanisms

- **protection (passive)**
  - diversity
  - hot-standby
  - usually no shared capacity

- **Restotration (active)**
  - typically hared capacity
  - flow restoration
    - unrestricted (theoretical)
    - restricted
    - single back-up paths
  - link restoration
  - released capacity

Robust network design problem DR-U: Benders' decomposition

- **situation dependent path lists**
  - $p=1,2,\dots,P_{ds}$
  - δ_{edps} path is identified by (d,p,s)

- **objective** minimize $\mathbf{F}(\mathbf{y}) = \Sigma_e \xi_e y_e$

- **constraints**
  - $\Sigma_p x_{dps} = h_{ds}$ d=1,2,…,D, s=1,2,…,S
  - $\Sigma_d \Sigma_p \delta_{edps} x_{dps} \le \alpha_{es} y_e$ e=1,2,…,E, s=1,2,…,S

Robust network design problem DR-U: Benders' decomposition (Section 10.3.1)

Proposition 10.1

Link capacity vector is globally feasible iff for each vector $\pi = (\pi_e \geq 0: e=1,2,\ldots,E)$ such that $\Sigma_e \pi_e = 1$ and for each situation s=1,2,…,S, the inequality

$$\Sigma_e \pi_e \alpha_{es} y_e \geq \Sigma_d \lambda_d(\pi) h_{ds}$$

holds, where $\lambda_d(\pi)$ is the length of the π–shortest path for demand d.

- Using this result we can get gradually get rid of the flow variables, and use only link capacities as variables.
- This is a sequential procedure involving a master problem and feasibility tests.

Benders' decomposition

- Step 1: Initialize Ω
- Step 2: Solve the Master Problem (in variables **y**)
  - ***minimize*** $\mathbf{F}(\mathbf{y}) = \Sigma_e \xi_e y_e$
  - ***subject to*** all inequalities from Ω and $\mathbf{y} \geq \mathbf{0}$
- Step 3: For each situation s=1,2,…,S solve the feasibility test (in variables p and l)
  - ***maximize*** $W(l,p) = \Sigma_d \lambda_d h_{ds} - \Sigma_e \pi_e \alpha_{es} y_e$
  - ***subject to*** $p \geq \mathbf{0}$

    $\Sigma_e \pi_e = 1$

    $\lambda_d \leq \Sigma_e \delta_{edp} \pi_e$ d=1,2,…,D, p=1,2,…,P_{ds}
- If optimal W* is > 0 then add the following inequality to Ω :
  - $(\pi_1^* \alpha_{1s})y_1 + (\pi_2^* \alpha_{2s})y_2 + \ldots + (\pi_E^* \alpha_{Es})y_E \geq \Sigma_d \lambda_d^* h_{ds}$.
- Step 4: If all feasibility tests are positive then STOP: current **y** is globally optimal. Otherwise go to Step 2.

invented for modular links

Robust network design problem DR-U: path generation

- **situation dependent path lists**
  - $p=1,2,\ldots,P_{ds}$
  - δ_{edps} path is identified by (d,p,s)

- **objective** minimize $\mathbf{F}(\mathbf{y}) = \Sigma_e \xi_e y_e$

- **constraints**
  - $\Sigma_p x_{dps} = h_{ds}$ $d=1,2,\ldots,D$, $s=1,2,\ldots,S$
  - $\Sigma_d \Sigma_p \delta_{edps} x_{dps} \leq \alpha_{es} y_e$ $e=1,2,\ldots,E$, $s=1,2,\ldots,S$

Path generation (Section 10.1.1)

- **$L(x,y;\lambda,\pi)$ =** $\Sigma_d \Sigma_s \lambda_{ds} h_{ds} + \Sigma_s \Sigma_d \Sigma_p (\Sigma_e \delta_{edps} \pi_{es} - \lambda_{ds}) x_{dps} + \Sigma_e (\xi_e - \Sigma_s \alpha_{es} \pi_{es}) y_e$

- ***maximize*** $W(\lambda,\pi) = \Sigma_d \Sigma_s \lambda_{ds} h_{ds}$

  subject to $\lambda_{ds} \le \Sigma_e \delta_{edps} \pi_{es}$ $\quad d=1,2,\dots,D,\ p=1,2,\dots,P_{ds},\ s=1,2,\dots,S$

  $\Sigma_s \alpha_{es} \pi_{es} = \xi_e$ $\quad e=1,2,\dots,E$

  $\alpha_{es} = 0 \rightarrow \pi_{es} = +\infty$ $\quad e=1,2,\dots,E,\ s=1,2,\dots,S$

  $\pi \ge \mathbf{0}$

- **If for at least one situation s we find a path shortest than λ_{ds}* then we can possibly improve the solution!**
- **The rate of possible improvement: the difference.**

Robust network design problem DR-R: restricted flow reallocation

- **variables**
  - x_{dps} flow realising demand d on path p in situation s,
  - y_e capacity of link e
- **objective** minimise $\mathbf{F}(\mathbf{y}) = \Sigma_e \xi_e y_e$

$$\theta_{dps} = \prod_{\{e:\ \delta_{edp} = 1\}} \alpha_{es}$$

- **constraints**
  - $\Sigma_p x_{dps} = h_{ds}$ d=1,2,…,D , s=0,1,…,S
  - $x_{dps} \geq \theta_{dps} x_{dp0}$ d=1,2,…,D , p=1,2,…,P_d , s=1,2,…,S
  - $\Sigma_d \Sigma_p \delta_{edp} x_{dps} \leq \alpha_{es} y_e$ e=1,2,…,E , s=0,1,…,S
  - all variables non-negative

Another formulation with less constraints can be used – think about it.

Robust network design problem DR-R: alternative formulation

- **variables**
  - x_{dps} flow realising demand d on path p in situation s,
  - y_e capacity of link e

- **objective** minimise $\mathbf{F}(\mathbf{y}) = \Sigma_e \xi_e y_e$

$$\theta_{dps} = \prod_{\{e:\ \delta edp = 1\}} \alpha_{es}$$

- **constraints**
  - $\Sigma_p x_{dp0} = h_d$ d=1,2,…,D
  - $\Sigma_d \Sigma_p \delta_{edp} x_{dp0} \leq y_e$ e=1,2,…,E
  - $\Sigma_p x_{dps} + \Sigma_p \theta_{dps} x_{dp0} \geq h_d$ d=1,2,…,D , s=1,2,…,S
  - $\Sigma_d \Sigma_p \delta_{edp} x_{dps} \leq \alpha_{es}(y_e - \Sigma_d \Sigma_p \delta_{edp} \theta_{dps} x_{dp0})$ e=1,2,…,E , s=1,2,…,S
  - all variables non-negative

Benders' decomposition for DR-R (Section 10.3.2)

- Step 1: Initialize Ω
- Step 2: Solve the Master Problem (in variables **y** and $\mathbf{x}_0$)
  - ***minimize*** $\mathbf{F}(\mathbf{y}) = \Sigma_e \xi_e y_e$
  - ***subject to*** all inequalities from Ω, $\mathbf{y},\mathbf{x}_0 \geq \mathbf{0}$, and
    - $\Sigma_p x_{dp0} = h_d$ $\quad$ $d=1,2,\dots,D$
    - $\Sigma_d \Sigma_p \delta_{edp} x_{dp0} \leq y_e$ $\quad$ $e=1,2,\dots,E$
- Step 3: For each situation s=1,2,…,S solve the feasibility test (in variables p and l)
  - ***maximize*** $W(l,p) = \Sigma_d \lambda_d(h_d - \Sigma_p \theta_{dps} x_{dp0}^*) - \Sigma_e \pi_e \alpha_{es}(y_e^* - \Sigma_d \Sigma_p \delta_{edp} \theta_{dps} x_{dp0}^*)$
  - ***subject to*** $p \geq \mathbf{0}$

    $\Sigma_e \pi_e = 1$

    $\lambda_d \leq \Sigma_e \delta_{edp} \pi_e$ $\quad$ $d=1,2,\dots,D,\ \ p=1,2,\dots,P_d$

  If optimal W* is > 0 then add the appropriate inequality to Ω:

  - $\Sigma_d \lambda_d^*(h_d - \Sigma_p \theta_{dps} x_{dp0}) - \Sigma_e \pi^* \alpha_{es}(y_e - \Sigma_d \Sigma_p \delta_{edp} \theta_{dps} x_{dp0}) \geq 0.$

- Step 4: If all feasibility tests are positive then STOP: current **y*** and $\mathbf{x}_0$* are globally optimal. Otherwise go to Step 2.

Path generation for DR-R (Section 10.1.2)

- For s =1,2,...,S the link metrics are just optimal dual variables π_{es}^*.

- For s =0, however, it is not that simple:

  - $\pi_e' = \Sigma_{s \in S(d,p) \cup \{0\}} \pi_{es}^*$

  where S(d,p) – situations for which path (d,p) works.

Can be copmuted easily for a given list, but not in general (no Dijkstra).

Robust network design problem DR-F: single normal path + backup path

- **indices**
  - $d=1,2,\ldots,D$ demands
  - $p=1,2,\ldots,P_d$ candidate pairs (P_{dp},Q_{dp}) of situation disjoint paths for demand d; P_{dp} - normal path, Q_{dp} – backup path
  - $e=1,2,\ldots,E$ links
  - $s=0,1,\ldots,S$ failure situations (0 - nominal state)
- **constants**
  - h_d volume of demand d (in all states)
  - ξ_e unit cost of link e
  - δ_{edp} = 1 if e belongs to normal path P_{dp}; 0, otherwise
  - β_{edp} = 1 if e belongs to backup path Q_{dp}; 0, otherwise
  - α_{es} link failure coefficient of link e in situation s ($\alpha_{es} \in \{0,1\}$)
  - θ_{dps} $= \prod_{\{e:\ \delta edp=1\}} \alpha_{es}$ (for binary α_{es})

Robust network design problem DR-F: single normal path + backup path

- **variables**
  - u_{dp} binary flow variable selecting pair (P_{dp}, Q_{dp})
  - y_e capacity of link e (continuous)
- **objective** minimize $\mathbf{F}(\mathbf{y}) = \Sigma_e \xi_e y_e$
- **constraints**
  - $\Sigma_p u_{dp} = 1$ d=1,2,…,D
  - $\Sigma_d \Sigma_p (\delta_{edp}\theta_{dps} + \beta_{edp}(1 - \theta_{dps}))\cdot h_d u_{dp} \le \alpha_{es} y_e$ e=1,2,…,E , s=0,1,…,S

$\theta_{dps} = \prod_{\{e:\ \delta edp = 1\}} \alpha_{es}$

shared protection capacity

Robust network design problem DR-HS: single normal path + HS backup path

- **variables**
  - u_{dp} binary flow variable selecting pair (P_{dp}, Q_{dp})
  - y_e capacity of link e (continuous)
- **objective** minimize $\mathbf{F}(\mathbf{y}) = \Sigma_e \xi_e y_e$
- **constraints**
  - $\Sigma_p u_{dp} = 1$ $\quad d=1,2,\ldots,D$
  - $\Sigma_d \Sigma_p (\delta_{edp} + \beta_{edp}) \cdot h_d u_{dp} \leq \alpha_{es} y_e$ $\quad e=1,2,\ldots,E , s=0,1,\ldots,S$

dedicated protection capacity

Link restoration - notation

- **indices**
  - $d=1,2,\ldots,D$ demands
  - $p=1,2,\ldots,P_d$ candidate paths for demand d
  - $e,l=1,2,\ldots,E$ links
  - $q=1,2,\ldots,Q_e$ candidate paths for restoring link e
- **constants**
  - h_d volume of demand d (in all states)
  - ξ_e unit cost of link e
  - δ_{edp} = 1 if e belongs to P_{dp}; 0, otherwise
  - β_{leq} = 1 if l belongs to path Q_{dp} restoring link e; 0, otherwise

Link restoration – problem formulation

- **variables**
  - x_{dp0} normal flow on path P_{dp}
  - y_e normal capacity of link e
  - z_{eq} flow restoring normal capacity of link e on path Q_{eq}
  - y_e' protection capacity of link e
- **objective** minimize $\mathbf{F}(\mathbf{y}) = \Sigma_e \xi_e (y_e + y_e')$
- **constraints**
  - $\Sigma_p x_{dp0} = h_d$ d=1,2,…,D
  - $\Sigma_d \Sigma_p \delta_{edp} x_{dp0} = y_e$ e=1,2,…,E
  - $\Sigma_q z_{eq} = y_e$ e=1,2,…,E
  - $\Sigma_q \beta_{leq} z_{eq} \leq y_l'$ e,l=1,2,…,E, $e \neq l$ (e – restored in l)

Other protection/restoration mechanisms and related design problems

Path protection (already discussed)

- shared spare capacity, reused released capacity
- situation-dependent backup paths
- single-path allocation

- Link protection
  - shared spare capacity, capacity not reused
  - capacity of a failed link is restored

- Hot-Standby
  - dedicated protection capacity

- Modular links, flows, single-path allocation

Chapter 9

Multi-layer networks

- Networks do have multiple layers of resources
  - traffic (demand) layer
  - trunk groups layer
  - transmission layer with sublayers
  - optical fibres layer
  - cable layer
  - duct layer

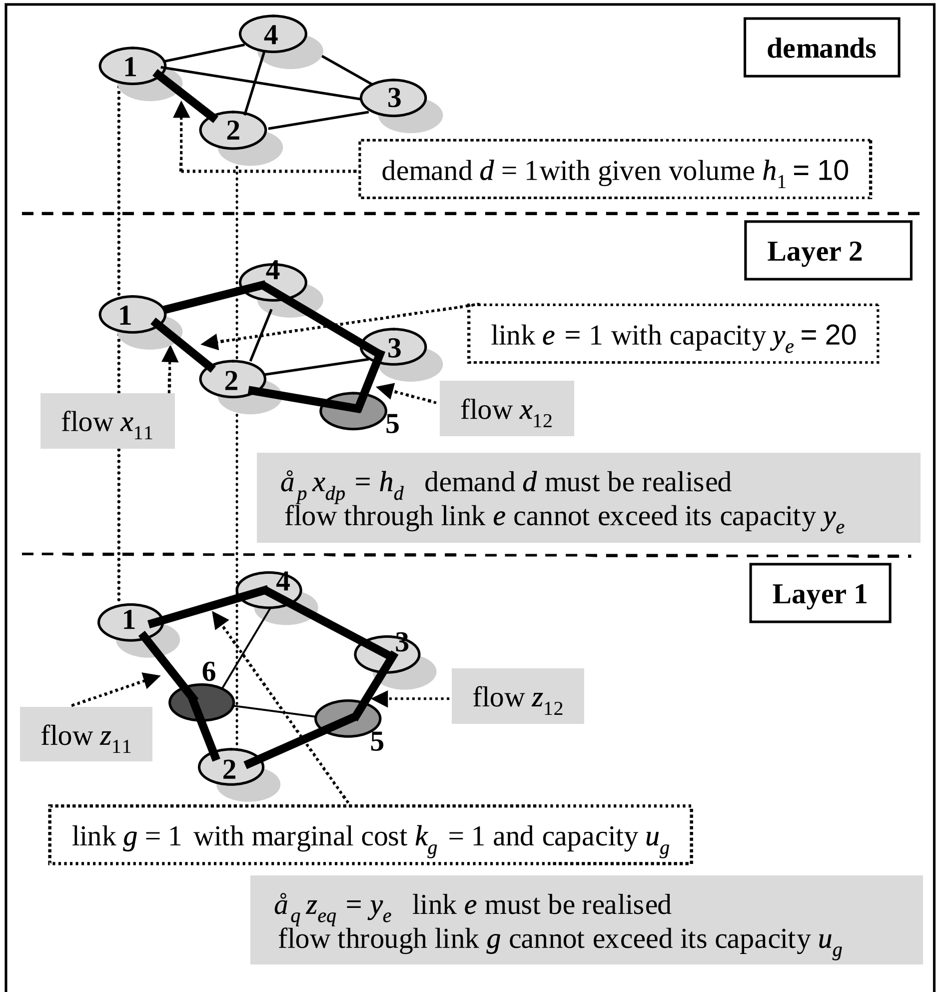
- Common rule
  - capacity of the links of the upper layer are demands for the lower layer below
  - these capacities are realized by means of path flows in the lower layer

Chapter 12

Two-layer design problem (TLDP)

- indices
  - $d=1,2,\ldots,D$ — demands
  - $p=1,2,\ldots,P_d$ — paths for flows realizing demand d
  - $e=1,2,\ldots,E$ — links in layer 2 (upper layer)
  - $q=1,2,\ldots,Q_e$ — paths for flows realizing capacity of link e
  - $g=1,2,\ldots,G$ — links in layer 1 (lower layer)
- constants
  - h_d — volume of demand d
  - κ_g — unit cost of link g
  - δ_{edp} — = 1 if e belongs to path p realising demand d; 0, otherwise
  - γ_{geq} — = 1 if g belongs to path q realising link e; 0, otherwise

Two-layer design problem (TLDP)



Two-layer network

Two-layer design problem (TLDP)

- **variables**
  - x_{dp} upper layer flow realizing demand d on path p
  - z_{eq} lower layer flow realizing link e on path q
  - y_e capacity of upper layer link e
  - u_g capacity of lower layer link g
- **objective** minimize $\mathbf{F}(\mathbf{u}) = \Sigma_g \kappa_g u_g$
- **constraints**
  - $\Sigma_p x_{dp} = h_d$ $d=1,2,\dots,D$
  - $\Sigma_d \Sigma_p \delta_{edp} x_{dp} = y_e$ $e=1,2,\dots,E$
  - $\Sigma_q z_{eq} = y_e$ $e=1,2,\dots,E$
  - $\Sigma_e \Sigma_q \gamma_{geq} z_{eq} \le u_g$ $g=1,2,\dots,G$
  - all variables non-negative

Two-layer design problem (TLDP) - solution

- **Shortest path allocation rules applies:**
  - Step 1: Compute the length ξ_e of the shortest path for each link e with respect to lower layer link costs κ_g; let Q_e be such a path.
  - Step 2: Compute the length λ_d of the shortest path for each demand d with respect to upper layer link costs ξ_e; let P_d be such a path.
  - Step 3: For each demand d allocate the whole demand volume h_d to path P_d. For each link e compute the resulting load y_e.
  - Step 4: For each link e allocate the whole link capacity y_e to path Q_e. For each link g compute the resulting load u_g.

- **Extensions:**
  - single-path allocation
  - modular links
  - modular flows
  - node-link formulation

$$\Sigma_e \Sigma_q \gamma_{geq} z_{eq} \le M u_g$$

Two-layer allocation problem (TLAP)

- **variables**
  - x_{dp} upper layer flow realizing demand d on path p
  - z_{eq} lower layer flow realizing link e on path q
  - y_e capacity of upper layer link e

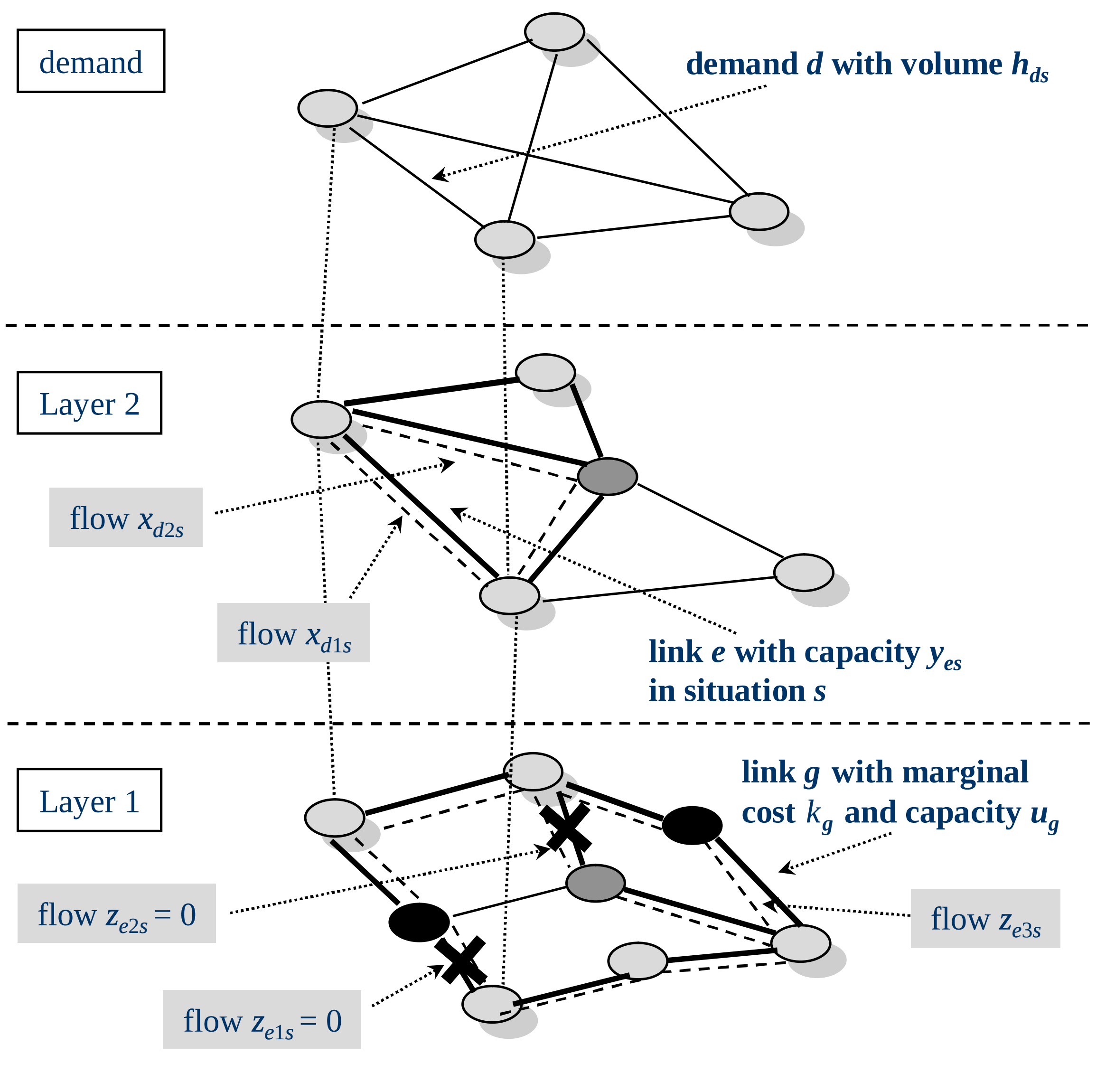
  (c_g capacity of lower layer link g – constant)

- **constraints**
  - $\Sigma_p x_{dp} = h_d$ d=1,2,…,D
  - $\Sigma_d \Sigma_p \delta_{edp} x_{dp} = y_e$ e=1,2,…,E
  - $\Sigma_q z_{eq} = y_e$ e=1,2,…,E
  - $\Sigma_e \Sigma_q \gamma_{geq} z_{eq} \leq c_g$ g=1,2,…,G
  - all variables non-negative

Two-layer robust design problem (TLDP-U)

- indices
  - $d=1,2,\ldots,D$ — demands
  - $p=1,2,\ldots,P_d$ — paths for flows realizing demand d
  - $e=1,2,\ldots,E$ — links in layer 2 (upper layer)
  - $q=1,2,\ldots,Q_e$ — paths for flows realizing capacity of link e
  - $g=1,2,\ldots,G$ — links in layer 1 (lower layer)
  - $s=0,1,\ldots,S$ — failure situations (0 - nominal state)
- constants
  - h_{ds} — volume of demand d to be realised in situation s
  - κ_g — unit cost of link g
  - δ_{edp} — = 1 if e belongs to path p realising demand d; 0, otherwise
  - γ_{geq} — = 1 if g belongs to path q realising link e; 0, otherwise
  - α_{gs} — binary link failure coefficient for link g in situation s

TLDP-U



Two-layer network with failures

Two-layer robust design problem (TLRDP-U)

- **variables**
  - x_{dps} upper layer flow realizing demand d on path p in situation s
  - z_{eqs} lower layer flow realizing link e on path q in situation s
  - y_{es} capacity of upper layer link e in situation s
  - u_g capacity of lower layer link g
- **objective** minimize $\mathbf{F}(\mathbf{u}) = \Sigma_g \kappa_g u_g$
- **constraints**
  - $\Sigma_p x_{dps} = h_{ds}$ d=1,2,…,D , s=0,1,…,S
  - $\Sigma_d \Sigma_p \delta_{edp} x_{dps} = y_{es}$ e=1,2,…,E , s=0,1,…,S
  - $\Sigma_q z_{eqs} = y_{es}$ e=1,2,…,E , s=0,1,…,S
  - $\Sigma_e \Sigma_q \gamma_{geq} z_{eqs} \leq \alpha_{gs} u_g$ g=1,2,…,G , s=0,1,…,S
  - all variables are non-negative

Two-layer robust design problem (TLRDP-LLR)

- **variables**
  - x_{dp} upper layer flow realizing demand d on path p
  - z_{eqs} lower layer flow realizing link e on path q in situation s
  - y_e capacity of upper layer link e
  - u_g capacity of lower layer link g
- **objective** minimize $\mathbf{F}(\mathbf{u}) = \Sigma_g \kappa_g u_g$
- **constraints**
  - $\Sigma_p \; x_{dp} = h_d$ d=1,2,…,D
  - $\Sigma_d \Sigma_p \delta_{edp} x_{dp} = y_e$ e=1,2,…,E
  - $\Sigma_q \; z_{eqs} = y_e$ e=1,2,…,E , s=0,1,…,S
  - $\Sigma_e \Sigma_q \gamma_{geq} z_{eqs} \leq \alpha_{gs} u_g$ g=1,2,…,G , s=0,1,…,S
  - all variables are non-negative

Two-layer robust design problem (TLRDP-ULR)

- **variables**
  - x_{dps} — upper layer flow realizing demand d on path p in situation s
  - z_{eq} — lower layer flow realizing link e on path q
  - y_{es} — capacity of upper layer link e in situation s
  - u_g — capacity of lower layer link g
- **objective** — minimize $\mathbf{F}(\mathbf{u}) = \Sigma_g \kappa_g u_g$
- **constraints**
  - $\Sigma_p x_{dps} = h_{ds}$ — $d=1,2,\dots,D$, $s=0,1,\dots,S$
  - $\Sigma_d \Sigma_p \delta_{edp} x_{dps} = y_{es}$ — $e=1,2,\dots,E$, $s=0,1,\dots,S$
  - $\Sigma_q \theta_{eqs} z_{eq} \geq y_{es}$ — $e=1,2,\dots,E$, $s=0,1,\dots,S$
  - $\Sigma_e \Sigma_q \gamma_{geq} z_{eq} \leq u_g$ — $g=1,2,\dots,G$
  - all variables are non-negative

$$\theta_{eqs} = \Pi_{\{g:\ \gamma geq = 1\}}\ \alpha_{gs}$$

Two-layer robust design problems: remarks, extensions, solutions

- **Networks can have many layers**
  - **IP over ATM over WDM over cable infrustracture (4 layers)**
- **Extensions**
  - restricted reconfiguration, single backup paths, hot-standby
  - single-path allocation
  - modular links, modular flows
  - node-link formulations
- **Solution methods**
  - LP, MIP, IP
  - iterative approximate
  - stochastic metaheuristics

- **Decomposition approach (different operators in different layers)**

Two-layer modular design problem (TLDP/M) - iterative approximate solution

- m Step 0: Perform the single path allocation with respect to $k = (\kappa_1,\kappa_2,..., \kappa_G)$. Compute the resulting lower layer link loads $\underline{u}_g$ and link capacities u_g, g=1,2,...,G.
- m Step 1: Compute the lower layer link metrics: $\rho_g := \kappa_g \times (u_g / \underline{u}_g)$, g=1,2,...,G.
- m Step 2: Compute the length ξ_e of the shortest path for each link e with respect to lower layer link costs r; let Q_e be such a path.
- m Step 3: Compute the length λ_d of the shortest path for each demand d with respect to upper layer link costs x; let P_d be such a path.
- m Step 4: For each demand d allocate the whole demand volume h_d to path P_d. For each link e compute the resulting load y_e.
- m Step 5: For each link e allocate the whole link capacity y_e to path Q_e. For each link g compute the resulting link load $\underline{u}_g$ and link capacity u_g.
- m Step 6: Coverged? If yes, stop. If not, go to Step 1.

$$\Sigma_e \Sigma_q \gamma_{geq} z_{eq} \le M u_g$$

Two-layer allocation problem (TLAP) - iterative approximate solution

- Step 0: Perform the single path allocation with respect to r = (1,1,...,1). Compute the resulting lower layer link loads $\underline{u}_g$ and link capacities u_g, g=1,2,...,G.
- Step 1: Compute the lower layer link metrics: $\rho_g := (\underline{u}_g / c_g)$, g=1,2,...,G.
- Step 2: Compute the length ξ_e of the shortest path for each link e with respect to lower layer link costs r; let Q_e be such a path.
- Step 3: Compute the length λ_d of the shortest path for each demand d with respect to upper layer link costs x; let P_d be such a path.
- Step 4: For each demand d allocate the whole demand volume h_d to path P_d. For each link e compute the resulting load y_e.
- Step 5: For each link e allocate the whole link capacity y_e to path Q_e. For each link g compute the resulting link load $\underline{u}_g$.
- Step 6: Coverged? If yes, stop. If not, go to Step 1.

$\Sigma_e \Sigma_q \gamma_{geq} z_{eq} \leq c_g$

Concluding remarks I

- Multicommodity flow approach to network modelling
  - captures most of important cases
  - traffic issues elimnated through the use of a proper demand matrix
  - powerful optimization methods developed
- Most of the problems are NP-hard
  - IP methods: branch and cut
  - decomposition
  - stochastic heusristics
    - evolutionary algorithms
    - simulated allocation
  - approximate methods (shortest path allocation)

Concluding remarks II

- New, challanging problems appear all the time
  - shortest-path routing
  - modelling of muti-layer networks (grooming, GMPLS)
  - enhancements of branch and cut
  - ...