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)
m To present basic approaches in network modeling and optimization
  n multi-commodity flow networks
  n routing/resilience/multi-layering
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


and tons of papers ...
CONTENTS


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

## Dimensioning problem - Link-Path Formulation

### Indices
- $e = 1, 2, \ldots, E$: links
- $d = 1, 2, \ldots, D$: demands
- $p = 1, 2, \ldots, P_d$: paths for flows realizing demand $d$
  
  ($P_d \subseteq \{1, 2, \ldots, E\}$ – subset of the set of links indices)

### Constants
- $\delta_{edp}$: $= 1$ if $e$ belongs to path $p$ realizing demand $d$; 0, otherwise
  
  ($\delta_{edp} = 1$ iff $e \in P_{dp}$)
- $h_d$: volume of demand $d$
- $\xi_e$: unit (marginal) cost of link $e$
- $M$: module size of the link capacity
Dimensioning problem – MIP

**variables**
- \( x_{dp} \) \hspace{2em} \text{continuous flow realizing demand } d \text{ on path } p
- \( y_e \) \hspace{2em} \text{integer capacity of link } e

**objective**
- minimize \( F(y) = \sum_e \xi_e y_e \)

**constraints**
- \( \sum_p x_{dp} = h_d \) \hspace{2em} d=1,2,\ldots,D
- \( \sum_d \sum_p \delta_{edp} x_{dp} \leq M y_e \) \hspace{2em} e=1,2,\ldots,E
- all variables are \textit{non-negative}

How to solve this NP-hard problem?
Simple dimensioning problem – LP relaxation

**m variables**

n \( x_{dp} \)  
flow realizing demand \( d \) on path \( p \)

n \( y_e \)  
capacity of link \( e \)

**m objective**  
minimize \( F(y) = \sum_e \xi_e y_e \)

**m constraints**

n \( \sum_p x_{dp} = h_d \quad d=1,2,\ldots,D \)

n \( \sum_d \sum_p \delta_{edp} x_{dp} \leq y_e \quad e=1,2,\ldots,E \)  

n all variables are *continuous and non-negative*

How to solve this problem?
LP relaxation - solution

**m constraints**

\[ \sum_p x_{dp} = h_d \quad d=1,2,\ldots,D \]
\[ \sum_d \sum_p \delta_{edp} x_{dp} = y_e \quad e=1,2,\ldots,E \]

**m**

\[ F = \sum_e \xi_e \sum_d \sum_p \delta_{edp} x_{dp} = \sum_d \sum_p (\sum_e \xi_e \delta_{edp}) x_{dp} = \sum_d \sum_p \kappa_{dp} x_{dp} = \sum_d \zeta_d h_d \]

where

\( \kappa_{dp} \) - cost of path \( p \) of demand \( d \)
\( \zeta_d \) – cost of the cheapest (shortest with respect to \( \xi_e \)) path of demand \( d \)

**m 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 \( e \)
- \( \delta_{edp} = 1 \) if \( e \) belongs to path \( p \) realizing demand \( d \); 0, otherwise
Capacitated flow allocation problem – Link-Path formulation

\[ m \text{ variables} \]
\[ n \ x_{dp} \quad \text{flow realizing demand d on path p} \]

\[ m \text{ constraints} \]
\[ n \ \sum_p x_{dp} = h_d \quad \text{d=1,2,…,D} \]
\[ n \ \sum_d \sum_p \delta_{edp} x_{dp} \leq c_e \quad \text{e=1,2,…,E} \]

n flow variables are \textit{continuous and non-negative}.

\[ \text{D+E property:} \]
\[ \text{If the problem is feasible then there exists a solution with at most D+E non-zero flows.} \]
**Node-link formulation**

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

**indices**
- \( d = 1, 2, \ldots, D \) demands
- \( v = 1, 2, \ldots, V \) nodes
- \( e = 1, 2, \ldots, E \) 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**

\[ \sum_e a_{ev} x_{ed} - \sum_e b_{ev} x_{ed} = h_d \quad \text{if} \ v = s_d \]
\[ = 0 \quad \text{if} \ v \neq s_d, t_d \]
\[ = -h_d \quad \text{if} \ v = t_d \]

\[ v = 1, 2, \ldots, V \quad d = 1, 2, \ldots, D \]

\[ \sum_d x_{ed} \leq c_e \]

\[ e = 1, 2, \ldots, E \]
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 \( w \)
- \( H_v = \sum_{w \neq v} h_{vw} \) 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**

**m variables**

\[ x_{ev} \geq 0 \]

flow realizing all demands originating at node \( v \)
on link \( e \)

**m constraints**

\[ \sum_e a_{ev} x_{ev} = H_v \quad \text{v=1,2,...,V} \quad \text{(redundant)} \]

\[ \sum_e a_{ew} x_{ev} - \sum_e b_{ew} x_{ev} = -h_{vw} \quad \text{v,w=1,2,...,V} \quad v \neq w \]

\[ \sum_v x_{ev} \leq c_e \quad \text{e=1,2,...,E} \]
<table>
<thead>
<tr>
<th></th>
<th>#variables</th>
<th>#constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>L-P</strong></td>
<td>$P \times V(V-1) = O(V^2)$</td>
<td>$V(V-1) + (k \times V)/2 = O(V^2)$</td>
</tr>
<tr>
<td><strong>N-L</strong></td>
<td>$(k \times V \times V(V-1))/2 = O(V^3)$</td>
<td>$V \times V(V-1) + (k \times V)/2 = O(V^3)$</td>
</tr>
<tr>
<td><strong>A/N-L</strong></td>
<td>$(k \times V \times V)/2 = O(V^2)$</td>
<td>$V \times V + (k \times V)/2 = O(V^2)$</td>
</tr>
</tbody>
</table>

**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 $\pi_e$ associated 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

m variables
n \( x_{dp} \) flow realizing demand d on path p

m constraints
n \( \sum_p x_{dp} = h_d \) \( d=1,2,\ldots,D \)

n \( \sum_d \sum_p \delta_{edp} x_{dp} \leq c_e \) \( e=1,2,\ldots,E \)

n flow variables are continuous and non-negative

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

m How to define the proper lists of paths? **By path generation.**
Capacitated flow allocation problem – adjusted formulation

m variables
n \( x_{dp} \) flow realizing demand \( d \) on path \( p \)
n \( z \) auxiliary variable

m objective
minimize \( z \)

m constraints
n \( \sum_p x_{dp} = h_d \) \( d=1,2,\ldots,D \) \( (\lambda_d \text{ - unconstrained}) \)
n \( \sum_d \sum_p \delta_{edp} x_{dp} \leq c_e + z \) \( e=1,2,\ldots,E \) \( (\pi_e \geq 0) \)

n flow variables are continuous and non-negative, \( z \) is continuous
Dual

\[ L(x,z; \pi, \lambda) = z + \sum_d \lambda_d (h_d - \sum_p x_{dp}) + \sum_e \pi_e (\sum_d \sum_p \delta_{edp} x_{dp} - c_e - z) \]

\[ x_{dp} \geq 0 \quad \text{for all} \ (d,p) \]

\[ W(\pi, \lambda) = \min_{x \geq 0, z} L(x,z; \pi, \lambda) \]

Dual

\[ \text{maximize} \ W(\pi, \lambda) = \sum_d \lambda_d h_d - \sum_e \pi_e c_e \]

subject to

- \[ \sum_e \pi_e = 1 \]
- \[ \lambda_d \leq \sum_e \delta_{edp} \pi_e \quad d=1,2,\ldots,D \quad p=1,2,\ldots,P_d \]
- \[ \pi_e \geq 0 \quad e=1,2,\ldots,E \]
Path generation - the reason

**Dual**

maximize $\sum_d \lambda_d h_d - \sum_e \pi_e c_e$

subject to

- $\sum_e \pi_e = 1$
- $\lambda_d \leq \sum_e \delta_{edp} \pi_e$ for $d=1,2,\ldots,D$ and $p=1,2,\ldots,P_d$
- $\pi_e \geq 0$ for $e=1,2,\ldots,E$

if we can find a path shorter than $\sum_e \delta_{edp} \pi_e^*$ 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 $\pi^*$.
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 the 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.
Path diversity (works for disjoint paths)

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

**Constraints**
- $\sum_p x_{dp} = h_d \quad d=1,2,\ldots,D$
- $x_{dp} \leq h_d / n_d \quad d=1,2,\ldots,D, \ p=1,2,\ldots,P_d$
- $\sum_d \sum_p \delta_{edp} x_{dp} = y_e \quad e=1,2,\ldots,E$
- 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

- $\sum_p x_{dp} = h_d$ for $d=1,2,\ldots,D$
- $\sum_p \delta_{edp} x_{dp} \leq h_d / n_d$ for $e=1,2,\ldots,E$, $d=1,2,\ldots,D$
- $\sum_d \sum_p \delta_{edp} x_{dp} = y_e$ for $e=1,2,\ldots,E$
- all variables non-negative

In node-link formulation:

- $x_{ed} \leq h_d / n_d$ for $e=1,2,\ldots,E$, $d=1,2,\ldots,D$
Single path allocation (non-bifurcated flows)

m variables
n $u_{dp}$ binary flow variable corresponding to demand $d$ and path $p$

m constraints
n $\sum_p u_{dp} = 1$  \hspace{1cm} d=1,2,\ldots,D
n $\sum_d \sum_p \delta_{edp}(h_d u_{dp}) \leq c_e$  \hspace{1cm} e=1,2,\ldots,E
n $u$ binary

In node-link formulation? NP-hard
Single-path allocation: N-L formulation

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

- **Constraints**
  - \( \sum_e a_{ev} u_{ed} - \sum_e b_{ev} u_{ed} = 1 \) if \( v = s_d \)
  - \( \sum_e a_{ev} u_{ed} - \sum_e b_{ev} u_{ed} = 0 \) if \( v \neq s_d, t_d \)
  - \( \sum_e a_{ev} u_{ed} - \sum_e b_{ev} u_{ed} = -1 \) if \( v = t_d \)

\( v=1,2,\ldots,V \quad d=1,2,\ldots,D \)

\( \sum_d h_d u_{ed} \leq c_e \) \( e=1,2,\ldots,E \)

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 difference 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

<table>
<thead>
<tr>
<th>minimize $F(y) = \sum_e \xi_e y_e$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sum_d \sum_p \delta_{edp} x_{dp} \leq M \cdot y_e$ \hspace{1cm} $e=1,2,...,E$ \hspace{1cm} $y_e$ \hspace{1cm} integer - MIP</td>
</tr>
<tr>
<td>$\sum_d \sum_p \delta_{edp} x_{dp} \leq M \cdot y_e$ \hspace{1cm} $e=1,2,...,E$ \hspace{1cm} $x_{dp}$, $y_e$ \hspace{1cm} integer – IP</td>
</tr>
</tbody>
</table>

### Multiple modules

<table>
<thead>
<tr>
<th>minimize $F(y) = \sum_e \sum_k \xi_{ek} y_{ek}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sum_d \sum_p \delta_{edp} x_{dp} \leq \sum_k M_k \cdot y_{ek}$ \hspace{1cm} $e=1,2,...,E$ \hspace{1cm} $y_{ek}$ \hspace{1cm} integer – MIP</td>
</tr>
</tbody>
</table>

### Incremental modular capacities

<table>
<thead>
<tr>
<th>minimize $F(y) = \sum_e \sum_{ek} \xi_{ek} u_{ek}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sum_d \sum_p \delta_{edp} x_{dp} \leq \sum_k m_k \cdot u_{ek}$ \hspace{1cm} $e=1,2,...,E$ \hspace{1cm} $u_{ek}$ \hspace{1cm} binary – MIP</td>
</tr>
<tr>
<td>$u_{e1} \geq u_{e2} \geq ... \geq u_{eK}$ \hspace{1cm} $e=1,2,...,E$</td>
</tr>
</tbody>
</table>
Dimensioning problem – convex and concave variants

**Objective**

minimize \( F(y) = \sum_e \xi_e f(y_e) \)

\[ y_e = \sum_d \sum_p \delta_{edp} x_{dp} \]

\( f(z) \) – 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)

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

**m objective** minimize \( \sum_e \xi_e y_e \)

**m constraints**
- \( n \ \sum_p x_{dp} = h_d \quad d=1,2,\ldots,D \)
- \( n \ \sum_d \sum_p \delta_{edp} x_{dp} \leq y_e \quad e=1,2,\ldots,E \)
Benders’ decomposition

Iterative process

- eliminate flow variables $x$
- introduce new constraints on $y$

Section 5.4.3
BD scheme

m Step 0
n Ω := \{ y_e \geq 0 : e=1,2,...,E \} \quad \text{(set of inequality constraints on y)}

m Step 1: solve master problem (y*)
n minimize \( \Sigma_e \xi_e y_e \)

n subject to \( \Omega \)

m Step 2: perform the feasibility test (\( \pi^*, \lambda^* \))

n maximize \( W(\pi,\lambda) = \Sigma_d \lambda_d h_d - \Sigma_e \pi_e y_e^* \)

n subject to

\( \Sigma_e \pi_e = 1 \)

\( \lambda_d \leq \Sigma_e \delta_{edp} \pi_e \quad d=1,2,...,D \quad p=1,2,...,P_d \)

\( \pi_e \geq 0 \quad e=1,2,...,E \)

if \( W(\pi^*,\lambda^*) > 0 \) then add inequality \( \Sigma_d \lambda_d^* h_d - \Sigma_e \pi_e^* y_e \leq 0 \) to \( \Omega \)

m 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**
    - \( \sum_{p} x_{dp} = h_d \quad \text{d}=1,2,\ldots,D \quad (\lambda_d \text{ - unconstrained}) \)
    - \( \sum_{d} \sum_{p} \delta_{edp} x_{dp} \leq y_{e}^{*} + z \quad \text{e}=1,2,\ldots,E \quad (\pi_e \geq 0) \)
    - 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

**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 $F = \sum_e \xi_e y_e + \sum_e \kappa_e u_e$

**constraints**

- $\sum_p x_{dp} = h_d$ \hspace{1cm} $d=1,2,\ldots,D$
- $\sum_d \sum_p \delta_{edp} x_{dp} = y_e$ \hspace{1cm} $e=1,2,\ldots,E$
- $y_e \leq M_e u_e$ \hspace{1cm} $e=1,2,\ldots,E$

- $y$ and $x$ non-negative, and $u$ binary

*NP-hard*
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 administrative weights (simplest $w_e = 1$)
  - $w_e$ weight (metric) of link $e$, $w = (w_1, w_2, \ldots, w_E)$.

- 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 $h_d$. 
Shortest Path Routing

**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 \( e \)
- \( \delta_{edp} = 1 \) if \( e \) belongs to path \( p \) realizing demand \( d \); 0, otherwise
Shortest Path Routing

**variables**

- \( w_e \) weight (metric) of link \( e \), \( w = (w_1, w_2, \ldots, w_E) \)
- \( x_{dp}(w) \) ECMP flow induced by metric system \( w \) on path \( (d, p) \)

**constraints**

- \( \sum_p x_{dp}(w) = h_d \) \quad \text{d}=1,2,\ldots,D
- \( \sum_d \sum_p \delta_{edp} x_{dp}(w) \leq c_e \) \quad \text{e}=1,2,\ldots,E \quad (*)
- \( w \in W \)

**NP-hard**

Penalty function can be used instead of (*)

©Michał Pióro
ECMP (Equal Cost Multi-Path) rule

ECMP (Equal-split rule)

m ECMP flow  

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

m all link weights $w_e$ equal to 1 

m ECMP flow  

n from s to t  

n from t to s
How to compute ECMP flows for given weights?

**indices**
- \( d = 1, 2, \ldots, D \) demands
- \( p = 1, 2, \ldots, P_d \) paths of demand \( d \)
- \( e = 1, 2, \ldots, E \) links
- \( s, t, v = 1, 2, \ldots, V \) 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**
  - \( r_d \): Length 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 \( v \)
m maximize \( E \sum_d r_d + \sum_e \sum_t z_{et} \)

m constraints

\( r_d \leq \sum_e \delta_{edp} w_e \quad \text{d=1,2,...,E, p=1,2,...,P_d} \)

\( z_{et} \leq \sum_e \delta_{edp} w_e - r_d \quad \text{d=1,2,...,E, p=1,2,...,P_d, t=j(d), e \in (d,p)} \)

\( 0 \leq z_{et} \leq 1 \quad \text{e=1,2,...,E, t=1,2,...,V} \)

\( \sum_{\{e: j(e)=t\}} x_{et} = \sum_{s \neq t} h_{st} \quad \text{t=1,2,...,V} \)

\( \sum_{\{e: i(e)=v\}} x_{et} - \sum_{\{e: j(e)=v\}} x_{et} = h_{vt} \quad \text{t,v=1,2,...,V, t\neq v} \)

\( 0 \leq y_{i(e)t} - x_{et} \leq z_{et} \sum_v h_{vt} \quad \text{t=1,2,...,V, e=1,2,...,E} \)

\( x_{et} \leq (1-z_{et})(\sum_v h_{vt}) \quad \text{t=1,2,...,V, e=1,2,...,E} \)

we assume that if two paths have different length then they differ by at least one
ECMP flows for given weight - N-L formulation

\[ m \text{ maximize } \quad V \sum_s \sum_t r_{st} + \sum_e \sum_t z_{et} \]

\[ m \text{ constraints } \]

\[ n \quad r_{st} \leq w_e + r_{j(e)t} \quad \text{s}=1,2,\ldots,V, \text{ t}=1,2,\ldots,V, \text{ i(e)} = s \]

\[ n \quad z_{et} \leq w_e + r_{j(e)t} - r_{i(e)t} \quad \text{s}=1,2,\ldots,V, \text{ t}=1,2,\ldots,V \]

\[ n \quad 0 \leq z_{et} \leq 1 \quad \text{e}=1,2,\ldots,E, \text{ t}=1,2,\ldots,V \]

\[ n \quad \sum\{e: j(e)=t\} x_{et} = \sum_{s\neq t} h_{st} \quad \text{t}=1,2,\ldots,V \]

\[ n \quad \sum\{e: i(e)=v\} x_{et} - \sum\{e: j(e)=v\} x_{et} = h_{vt} \quad \text{t,v}=1,2,\ldots,V, \text{ t}\neq v \]

\[ n \quad 0 \leq y_{i(e)t} - x_{et} \leq z_{et} \sum_v h_{vt} \quad \text{t}=1,2,\ldots,V, \text{ e}=1,2,\ldots,E \]

\[ n \quad x_{et} \leq (1-z_{et})(\sum_v h_{vt}) \quad \text{t}=1,2,\ldots,V, \text{ e}=1,2,\ldots,E \]
Shortest Path Routing – MIP aggregated node-link formulation

Indices

- \( t, v, s = 1, 2, \ldots, V \) nodes
- \( e = 1, 2, \ldots, E \) 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

**m variables**

- \( w_e \) weight (metric) of link \( e \), \( w = (w_1, w_2, \ldots, w_E) \)
- \( r_{vt} \) 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 \( t \) \( (u_{et} = 1 - z_{et}) \)

**m no objective**

(can be added – e.g., minimization of maximal link utilization)
Shortest Path Routing – MIP formulation

\[ \sum_{e: j(e)=t} x_{et} = \sum_{s \neq t} h_{st} \quad t=1,2,...,V \]
\[ \sum_{e: i(e)=v} x_{et} - \sum_{e: j(e)=v} x_{et} = h_{vt} \quad t,v=1,2,...,V, t \neq v \]
\[ \sum_t x_{et} \leq c_e \quad e=1,2,...,E \]
\[ 0 \leq y_{i(e)t} - x_{et} \leq (1 - u_{et}) \sum_v h_{vt} \quad t=1,2,...,V, e=1,2,...,E \]
\[ x_{et} \leq u_{et} \sum_v h_{vt} \quad \]
\[ 0 \leq r_{j(e)t} + w_e - r_{i(e)t} \leq (1 - u_{et}) M \quad t=1,2,...,V, e=1,2,...,E \]
\[ 1 - u_{et} \leq r_{j(e)t} + w_e - r_{i(e)t} \quad t=1,2,...,V, e=1,2,...,E \]
\[ w_e \geq 1 \quad e=1,2,...,E \]
\[ w \in W \]
**MIP formulation - extensions**

**limited split**

\[ \sum_{\{e: i(e)=t\}} u_{et} \leq n \quad t, v = 1, 2, \ldots, V, \ t \neq v \]

**auxiliary objective function**

\[ \text{minimize} \quad t \]

\[ \text{subject to} \quad t \geq \left( \sum_t x_{et} \right) / c_e \quad e = 1, 2, \ldots, E \]

**dimensioning problem**

\[ \text{minimize} \quad F = \sum_e \xi_e y_e \]

\[ \text{use variables} \ y_e \text{ instead of} \ c_e \]

**installation costs** \( \kappa_e \) – 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 approaches

- 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)
Simulated Annealing – neighbourhood for ECMP

**ECMP Optimisation Problem**

solution space: $W$

evaluation function: maximal overload $F(w) = \max \{y_e(w)/c_e : e=1,2,...,E\}$
find $w \in W$ with minimizing $F(w)$

$N(w) \subseteq W$ – 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, \( \geq 1 \)) weights generating the paths used in the solution of Phase 1 (easy to find using an LP when exist) and convert them to integers.
Dual approach

m Skip weights and find single path routing:

variables

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

maximize \( F = \sum_e (c_e - \sum_d \sum_p \delta_{edp} x_{dp}) \)

constraints

\( \sum_p u_{dp} = 1 \quad d=1,2,\ldots,D \)
\( \sum_d \sum_p \delta_{edp} x_{dp} \leq c_e \quad e=1,2,\ldots,E \quad (\pi_e) \)

m Solve the dual problem and use weights \( w_e = 1 + \pi_e \).
Dual approach – dual problem

\textbf{m variables}
\begin{itemize}
  \item n \( \lambda_d \) unrestricted in sign
  \item n \( \pi_e \) non-negative
\end{itemize}

\textbf{m objective}

\[
\text{maximize } G = \sum_d \lambda_d h_d - \sum_e (1 + \pi_e) c_e
\]

\textbf{m constraints}

\[
\lambda_d \leq \sum_e \delta_{edp} (1 + \pi_e) \quad d=1,2,\ldots,E, \ p=1,2,\ldots,P_d
\]

When optimal \( \pi \) determine single paths for all demands we are done!

\[
w_e = 1 + \pi_e \text{ are the optimal weights (can be easily made integer)}
\]
Two-phase approach for single shortest paths

Phase 1:
Solve capacitated single-path allocation problem (MIP).

Phase 2:
Find (continuous, \( \geq 1 \)) 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

\[ \sum_p u_{dp} = 1 \quad d=1,2,\ldots,D \]

\[ \sum_d h_d \sum_p \delta_{edp} u_{dp} \leq c_e \quad e=1,2,\ldots,E \]
m It can happen that Phase 1 results in infeasible paths.

m 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).
Phase 2 – inverse shortest path problem

**indices**
- \( d = 1, 2, \ldots, D \)
- \( p = 0, 1, \ldots, P_d \) (p=0 path that is supposed to be the unique shortest path)
- \( e = 1, 2, \ldots, E \)

**variables**
- \( w_e \) weights

**constraints**
- \( \sum_e \delta_{ed0} w_e + 1 \leq \sum_e \delta_{edp} w_e \) \( d = 1, 2, \ldots, D \), \( p = 1, 2, \ldots, P_d \)
- \( w_e \geq 1 \) \( e = 1, 2, \ldots, E \)

Simple generation of two shortest paths applied to account for all paths
- If there are paths that make the 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:

$$\Sigma_{Pdp \in C} u_{dp} \leq 1$$

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

$$\sum_{p \in C} u_{dp} \leq |C| - 1$$

to the problem of Phase 1

and iterate

It is well known that there may be a set of $n$ paths with any $(n-1)$-element subset realizable (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 than 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
Resilient design: indices and constants

**Indices**
- \( d=1,2,...,D \): demands
- \( p=1,2,...,P_d \): paths for flows realizing demand \( d \)
- \( e=1,2,...,E \): links
- \( s=0,1,...,S \): failure situations (0 - nominal state)

**Constants**
- \( h_{ds} \): volume of demand \( d \) to be realized in situation \( s \)
- \( \xi_{e} \): unit cost of link \( e \)
- \( \delta_{edp} \): \( = 1 \) if \( e \) belongs to path \( p \) realising demand \( d \); 0, otherwise
- \( \alpha_{es} \): link failure coefficient of link \( e \) in situation \( s \)
  - (binary: \( \alpha_{es} \in \{0,1\} \), or fractional: \( 0 \leq \alpha_{es} \leq 1 \))
- \( \theta_{dps} \): \( = \prod_{\{e: \delta_{edp}=1\}} \alpha_{es} = \min\{ \alpha_{es}: e \in P_{dp} \} \) (for binary \( \alpha_{es} \))
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**

- \( \minimize F(y) = \sum_e \xi_e y_e \)

**Constraints**

- \( \sum_p \theta_{dps} x_{dp0} \geq h_{ds} \) for \( d=1,2,...,D \) and \( s=0,1,...,S \)
- \( \sum_d \sum_p \delta_{edp} x_{dp0} \leq y_e \) for \( e=1,2,...,E \)

- 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**

\[
\min \ F(y) = \sum_e \xi_e y_e
\]

**Constraints**

- \( \sum_p x_{dps} = h_{ds} \quad d=1,2,...,D, \ s=0,1,...,S \)
- \( \sum_d \sum_p \delta_{edp} x_{dps} \leq \alpha_{es} y_e \quad e=1,2,...,E, \ s=0,1,...,S \)
- 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.

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

Equipment

Capacities and flows

Minimal cost $F(y) = 4$

(non-bifurcated flows: $F(y) = 5$)
Another issue - cntd.

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

Layer 1

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

\[
\begin{align*}
\text{Layer 1} \\
(2) & \quad (1) \\
1 & \quad 1 & \quad 1 \\
\text{demand } h_{d1} &= 3 \\
\text{s} &= 1 \\
(2) & \quad (2) \\
1 & \quad 2 & \quad 1 \\
\text{demand } h_{d2} &= 3 \\
\text{s} &= 2 \\
(2) & \quad (2) \\
2 & \quad 0 & \quad 1 \\
\text{demand } h_{d3} &= 3 \\
\text{s} &= 3 \\
(2) & \quad (1) \\
2 & \quad 1 & \quad 0 \\
\text{demand } d_{d4} &= 3 \\
\text{s} &= 4 \\
\end{align*}
\]

integer flows; for continuous flows cost = 4.5
Protection/restoration mechanisms

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

- **Restoration (active)**
  - typically shared capacity
  - flow restoration
    - unrestricted (theoretical)
    - restricted
      - single back-up paths
  - link restoration
  - released capacity
Robust network design problem DR-U:
Benders’ decomposition

m situation dependent path lists
n \( p=1,2,\ldots,P_{ds} \)
\n\( \delta_{edps} \) path is identified by \((d,p,s)\)

m objective
minimize \( F(y) = \sum_e \xi_e y_e \)

m constraints
\n\( \sum_p x_{dps} = h_{ds} \quad d=1,2,\ldots,D, \ s=1,2,\ldots,S \)
\n\( \sum_d \sum_p \delta_{edps} x_{dps} \leq \alpha_{es} y_e \quad e=1,2,\ldots,E, \ s=1,2,\ldots,S \)
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 $\sum_e \pi_e = 1$ and for each situation $s=1,2,\ldots,S$, the inequality

$$\sum_e \pi_e \alpha_{es} y_e \geq \sum_d \lambda_d(\pi) h_{ds}$$

holds, where $\lambda_d(\pi)$ is the length of the $\pi$–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 \( \Omega \)

Step 2: Solve the Master Problem (in variables \( y \))

\[
\text{minimize} \quad F(y) = \sum_e \xi_e y_e \\
\text{subject to} \quad \text{all inequalities from } \Omega \text{ and } y \geq 0
\]

Step 3: For each situation \( s=1,2,\ldots,S \) solve the feasibility test (in variables \( \pi \) and \( \lambda \))

\[
\text{maximize} \quad W(\lambda,\pi) = \sum_d \lambda_d h_{ds} - \sum_e \pi_e \alpha_{es} y_e \\
\text{subject to} \quad \pi \geq 0 \\
\quad \sum_e \pi_e = 1 \\
\quad \lambda_d \leq \sum_e \delta_{edp} \pi_e \\
\text{d=1,2,\ldots,D, p=1,2,\ldots,P}_{ds}
\]

- If optimal \( W^* > 0 \) then add the following inequality to \( \Omega \):

\[
(\pi_1^* \alpha_{1s}) y_1 + (\pi_2^* \alpha_{2s}) y_2 + \ldots + (\pi_E^* \alpha_{Es}) y_E \geq \sum_d \lambda_d^* h_{ds}.
\]

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**
- \( p = 1, 2, \ldots, P_{ds} \)
- \( \delta_{edps} \) path is identified by \((d,p,s)\)

**objective** minimize \( F(y) = \sum_{e} \xi_{e} y_{e} \)

**constraints**
- \( \sum_{p} x_{dps} = h_{ds} \quad d = 1, 2, \ldots, D, \ s = 1, 2, \ldots, S \)
- \( \sum_{d} \sum_{p} \delta_{edps} x_{dps} \leq \alpha_{es} y_{e} \quad e = 1, 2, \ldots, E, \ s = 1, 2, \ldots, S \)
Path generation (Section 10.1.1)

\[ L(x,y;\lambda,\pi) = \sum_d \sum_s \lambda_{ds} h_{ds} + \sum_s \sum_d \sum_p (\sum_e \delta_{edps} \pi_{es} - \lambda_{ds}) x_{dps} + \sum_e (\xi_e - \sum_s \alpha_{es} \pi_{es}) y_e \]

**maximize** \[ W(\lambda,\pi) = \sum_d \sum_s \lambda_{ds} h_{ds} \]

**subject to**

\[ \lambda_{ds} \leq \sum_e \delta_{edps} \pi_{es} \] \( d=1,2,...,D \), \( p=1,2,...,P_{ds} \), \( s=1,2,...,S \)

\[ \sum_s \alpha_{es} \pi_{es} = \xi_e \] \( e=1,2,...,E \)

\[ \alpha_{es} = 0 \rightarrow \pi_{es} = +\infty \] \( e=1,2,...,E \), \( s=1,2,...,S \)

\[ \pi \geq 0 \]

**If for at least one situation s we find a path shortest than \( \lambda_{ds}^* \) 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 \( F(y) = \sum_e \xi_e y_e \)

\[ \theta_{dps} = \prod \{ e : \delta_{edp} = 1 \} \alpha_{es} \]

**constraints**

- \( \sum_p x_{dps} = h_{ds} \) \( d=1,2,...,D \), \( s=0,1,...,S \)
- \( x_{dps} \geq \theta_{dps} x_{dp0} \) \( d=1,2,...,D \), \( p=1,2,...,P_d \), \( s=1,2,...,S \)
- \( \sum_d \sum_p \delta_{edp} x_{dps} \leq \alpha_{es} y_e \) \( e=1,2,...,E \), \( s=0,1,...,S \)
- all variables non-negative

Another formulation with less constraints can be used – think about it.
Robust network design problem DR-R: alternative formulation

**m variables**

- $x_{dps}$: flow realising demand $d$ on path $p$ in situation $s$.
- $y_e$: capacity of link $e$.

**m objective**

minimise $F(y) = \sum_e \xi_e y_e$

**m constraints**

- $\sum_p x_{dp0} = h_d$  \hspace{1cm} d=1,2,...,D
- $\sum_d \sum_p \delta_{edp} x_{dp0} \leq y_e$  \hspace{1cm} e=1,2,...,E
- $\sum_p x_{dps} + \sum_p \theta_{dps} x_{dp0} \geq h_d$  \hspace{1cm} d=1,2,...,D, s=1,2,...,S
- $\sum_d \sum_p \delta_{edp} x_{dps} \leq \alpha_{es} (y_e - \sum_d \sum_p \delta_{edp} \theta_{dps} x_{dp0})$  \hspace{1cm} e=1,2,...,E, s=1,2,...,S
- all variables non-negative

$\theta_{dps} = \prod_{\{e: \delta_{edp}=1\}} \alpha_{es}$
Benders’ decomposition for DR-R (Section 10.3.2)

m Step 1: Initialize $\Omega$

m Step 2: Solve the Master Problem (in variables $y$ and $x_0$)

n \textbf{minimize} \quad F(y) = \sum_e \xi_e y_e

n \textbf{subject to} \quad \text{all inequalities from } \Omega, \quad y, x_0 \geq 0, \text{ and}

\quad \Sigma_p x_{dp0} = h_d \quad \text{d}=1,2,\ldots,D

\quad \Sigma_d \Sigma_p \delta_{edp} x_{dp0} \leq y_e \quad \text{e}=1,2,\ldots,E

m Step 3: For each situation s=1,2,\ldots,S solve the feasibility test (in variables $\pi$ and $\lambda$)

n \textbf{maximize} \quad W(\lambda,\pi) = \sum_d \lambda_d (h_d - \sum_p \theta_{dps} x_{dp0}^*) - \sum_e \pi_e \alpha_{es} (y_e^* - \sum_d \sum_p \delta_{edp} \theta_{dps} x_{dp0}^*)

n \textbf{subject to} \quad \pi \geq 0

\quad \sum_e \pi_e = 1

\quad \lambda_d \leq \sum_e \delta_{edp} \pi_e \quad \text{d}=1,2,\ldots,D, \quad p=1,2,\ldots,P_d

If optimal $W^*$ is > 0 then add the appropriate inequality to $\Omega$:

\quad \Sigma_d \lambda_d^* (h_d - \sum_p \theta_{dps} x_{dp0}) - \sum_e \pi^* \alpha_{es} (y_e - \sum_d \sum_p \delta_{edp} \theta_{dps} x_{dp0}) \geq 0.

m Step 4: If all feasibility tests are positive then STOP: current $y^*$ and $x_0^*$ are globally optimal. Otherwise go to Step 2.
For $s = 1, 2, ..., S$ the link metrics are just optimal dual variables $\pi_{es}^*$. 

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

$$\pi'_e = \sum_{s \in S(d,p) \cup \{0\}} \pi_{es}^*$$

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

Can be computed easily for a given list, but not in general (no Dijkstra).
## Robust network design problem DR-F: single normal path + backup path

<table>
<thead>
<tr>
<th>Indices</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d = 1, 2, \ldots, D )</td>
</tr>
<tr>
<td>( p = 1, 2, \ldots, P_d )</td>
</tr>
<tr>
<td>( e = 1, 2, \ldots, E )</td>
</tr>
<tr>
<td>( s = 0, 1, \ldots, S )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Constants</th>
</tr>
</thead>
<tbody>
<tr>
<td>( h_d )</td>
</tr>
<tr>
<td>( \xi_e )</td>
</tr>
<tr>
<td>( \delta_{edp} )</td>
</tr>
<tr>
<td>( \beta_{edp} )</td>
</tr>
<tr>
<td>( \alpha_{es} )</td>
</tr>
<tr>
<td>( \theta_{dps} )</td>
</tr>
</tbody>
</table>
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 $F(y) = \sum_e \xi_e y_e$

- **Constraints**
  - $\sum_p u_{dp} = 1$, $d=1,2,...,D$
  - $\sum_d \sum_p (\delta_{edp} \theta_{dps} + \beta_{edp}(1 - \theta_{dps})) \cdot h_d u_{dp} \leq \alpha_{es} y_e$, $e=1,2,...,E$, $s=0,1,...,S$

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

**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 \( F(y) = \sum_e \xi_e y_e \)

**Constraints**

\[
\begin{align*}
\sum_p u_{dp} &= 1, \quad d=1,2,\ldots,D \\
\sum_d \sum_p (\delta_{edp} + \beta_{edp}) \cdot h_d u_{dp} &\leq \alpha_{es} y_e, \quad e=1,2,\ldots,E, \quad s=0,1,\ldots,S
\end{align*}
\]
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)
- \( \xi_e \) unit cost of link \( e \)
- \( \delta_{edp} = 1 \) if \( e \) belongs to \( P_{dp} \); 0, otherwise
- \( \beta_{leq} = 1 \) 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**

$$\text{minimize } F(y) = \sum_e \xi_e (y_e + y_e')$$

**constraints**

- $\sum_p x_{dp0} = h_d$, $d=1,2,...,D$
- $\sum_d \sum_p \delta_{edp} x_{dp0} = y_e$, $e=1,2,...,E$
- $\sum_q z_{eq} = y_e$, $e=1,2,...,E$
- $\sum_q \beta_{leq} z_{eq} \leq y_{l'}$, $e,l=1,2,...,E$, $e \neq l$ (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
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
Two-layer design problem (TLDP)

Indices:
- $d=1,2,...,D$ demands
- $p=1,2,...,P_d$ paths for flows realizing demand $d$
- $e=1,2,...,E$ links in layer 2 (upper layer)
- $q=1,2,...,Q_e$ paths for flows realizing capacity of link $e$
- $g=1,2,...,G$ links in layer 1 (lower layer)

Constants:
- $h_d$ volume of demand $d$
- $\kappa_g$ unit cost of link $g$
- $\delta_{edp} = 1$ if $e$ belongs to path $p$ realizing demand $d$; 0, otherwise
- $\gamma_{geq} = 1$ if $g$ belongs to path $q$ realizing link $e$; 0, otherwise
Two-layer design problem (TLDP)

Two-layer network

Demand $d = 1$ with given volume $h_1 = 10$

Link $e = 1$ with capacity $y_e = 20$

Layer 1

Layer 2

Demand $d = 1$ with given volume $h_1 = 10$

Link $e = 1$ with capacity $y_e = 20$

Layer 1

Layer 2

$\sum p x_{dp} = h_d$ demand $d$ must be realised

Flow through link $e$ cannot exceed its capacity $y_e$

Link $g = 1$ with marginal cost $k_g = 1$ and capacity $u_g$

$\sum q z_{eq} = y_e$ link $e$ must be realised

Flow through link $g$ cannot exceed its capacity $u_g$
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 \( F(u) = \sum_g \kappa_g u_g \)

**constraints**

- \( \sum_p x_{dp} = h_d \quad d=1,2,\ldots,D \)
- \( \sum_d \sum_p \delta_{edp} x_{dp} = y_e \quad e=1,2,\ldots,E \)
- \( \sum_q z_{eq} = y_e \quad e=1,2,\ldots,E \)
- \( \sum_e \sum_q \gamma_{geq} z_{eq} \leq u_g \quad g=1,2,\ldots,G \)
- all variables non-negative
Two-layer design problem (TLDP) - solution

**Shortest path allocation rules applies:**

- **Step 1:** Compute the length $\xi_e$ of the shortest path for each link $e$ with respect to lower layer link costs $\kappa_g$; let $Q_e$ be such a path.
- **Step 2:** Compute the length $\lambda_d$ of the shortest path for each demand $d$ with respect to upper layer link costs $\xi_e$; 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

$$\sum_e \sum_q \gamma_{geq} z_{eq} \leq Mu_g$$
Two-layer allocation problem (TLAP)

**m variables**

- \( n \) \( x_{dp} \) upper layer flow realizing demand \( d \) on path \( p \)
- \( n \) \( z_{eq} \) lower layer flow realizing link \( e \) on path \( q \)
- \( n \) \( y_e \) capacity of upper layer link \( e \)

\((c_g \text{ capacity of lower layer link } g – \text{ constant})\)

**m constraints**

- \( n \) \( \sum_p x_{dp} = h_d \) \( d=1,2,...,D \)
- \( n \) \( \sum_d \sum_p \delta_{edp} x_{dp} = y_e \) \( e=1,2,...,E \)
- \( n \) \( \sum_q z_{eq} = y_e \) \( e=1,2,...,E \)
- \( n \) \( \sum_e \sum_q \gamma_{geq} z_{eq} \leq c_g \) \( g=1,2,...,G \)

\( n \) 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 realized in situation \( s \)
- \( \kappa_g \): unit cost of link \( g \)
- \( \delta_{edp} \): \( = 1 \) if \( e \) belongs to path \( p \) realizing demand \( d \); 0, otherwise
- \( \gamma_{geq} \): \( = 1 \) if \( g \) belongs to path \( q \) realizing link \( e \); 0, otherwise
- \( \alpha_{gs} \): binary link failure coefficient for link \( g \) in situation \( s \)
Two-layer network with failures

Layer 1
- flow $z_{e1s} = 0$
- flow $z_{e2s} = 0$
- link $g$ with marginal cost $\kappa_g$ and capacity $u_g$

Layer 2
- flow $x_{d1s}$
- flow $x_{d2s}$
- link $e$ with capacity $y_es$
- demand $d$ with volume $h_{ds}$

Demand

TLDP-U
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**
- \[ \text{minimize } F(u) = \sum_{g} \kappa_{g} u_{g} \]

**Constraints**
- \[ \sum_{p} x_{dps} = h_{ds} \quad d=1,2,\ldots,D, \ s=0,1,\ldots,S \]
- \[ \sum_{d} \sum_{p} \delta_{edp} x_{dps} = y_{es} \quad e=1,2,\ldots,E, \ s=0,1,\ldots,S \]
- \[ \sum_{q} z_{eqs} = y_{es} \quad e=1,2,\ldots,E, \ s=0,1,\ldots,S \]
- \[ \sum_{e} \sum_{q} \gamma_{eqz_{eqs}} \leq \alpha_{gs} u_{g} \quad g=1,2,\ldots,G, \ s=0,1,\ldots,S \]
- All variables are non-negative
Two-layer robust design problem (TLRDP-LLR)

**m 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 $g$

**m objective**
minimize $F(u) = \sum_g \kappa_g u_g$

**m constraints**
- $\sum_p x_{dp} = h_d \quad d = 1,2,\ldots,D$
- $\sum_d \sum_p \delta_{edp} x_{dp} = y_e \quad e = 1,2,\ldots,E$
- $\sum_q z_{eqs} = y_e \quad e = 1,2,\ldots,E, s = 0,1,\ldots,S$
- $\sum_e \sum_q \gamma_{geq} z_{eqs} \leq \alpha_{gs} u_g \quad g = 1,2,\ldots,G, s = 0,1,\ldots,S$
- 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 \( F(u) = \sum_g \kappa_g u_g \)

**Constraints**

- \( \sum_p x_{dps} = h_{ds} \) \( d=1,2,...,D \), \( s=0,1,...,S \)
- \( \sum_d \sum_p \delta_{edp} x_{dps} = y_{es} \) \( e=1,2,...,E \), \( s=0,1,...,S \)
- \( \sum_q \theta_{eqs} z_{eq} \geq y_{es} \) \( e=1,2,...,E \), \( s=0,1,...,S \)
- \( \sum_e \sum_q \gamma_{geq} z_{eq} \leq u_g \) \( g=1,2,...,G \)
- All variables are non-negative

\[ \theta_{eqs} = \prod_{\{g: \gamma_{geq} = 1\}} \alpha_{gs} \]
Two-layer robust design problems: remarks, extensions, solutions

- Networks can have many layers
  - IP over ATM over WDM over cable infrastructure (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 
\( \kappa = (\kappa_1, \kappa_2, \ldots, \kappa_G) \). Compute the resulting lower layer link loads \( u_g \) and link capacities \( u_g \), \( g=1,2,\ldots,G \).

m Step 1: Compute the lower layer link metrics: 
\[ \rho_g := \kappa_g \times \left( \frac{u_g}{u_g} \right), \quad g=1,2,\ldots,G. \]

m Step 2: Compute the length \( \xi_e \) of the shortest path for each link \( e \) with respect to lower layer link costs \( \rho \); let \( Q_e \) be such a path.

m Step 3: Compute the length \( \lambda_d \) of the shortest path for each demand \( d \) with respect to upper layer link costs \( \xi \); 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 \( u_g \) and link capacity \( u_g \).

m Step 6: Covered? If yes, stop. If not, go to Step 1.

\[ \Sigma_e \Sigma_q \gamma_{eq} z_{eq} \leq Mu_g \]
### Two-layer allocation problem (TLAP) - iterative approximate solution

- **Step 0:** Perform the single path allocation with respect to \( \rho = (1,1,...,1) \). Compute the resulting lower layer link loads \( u_g \) and link capacities \( u_g, g=1,2,...,G \).

- **Step 1:** Compute the lower layer link metrics:
  \[
  \rho_g := \left( \frac{u_g}{c_g} \right), \quad g=1,2,...,G.
  \]

- **Step 2:** Compute the length \( \xi_e \) of the shortest path for each link \( e \) with respect to lower layer link costs \( \rho \); let \( Q_e \) be such a path.

- **Step 3:** Compute the length \( \lambda_d \) of the shortest path for each demand \( d \) with respect to upper layer link costs \( \xi \); 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 \( u_g \).

- **Step 6:** Converged? If yes, stop. If not, go to Step 1.

\[
\sum_e \sum_q \gamma_{eq} z_{eq} \leq c_g
\]
Multicommodity flow approach to network modelling
- captures most of important cases
- traffic issues eliminated 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 heuristics
  - evolutionary algorithms
  - simulated allocation
- approximate methods (shortest path allocation)
Concluding remarks II

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