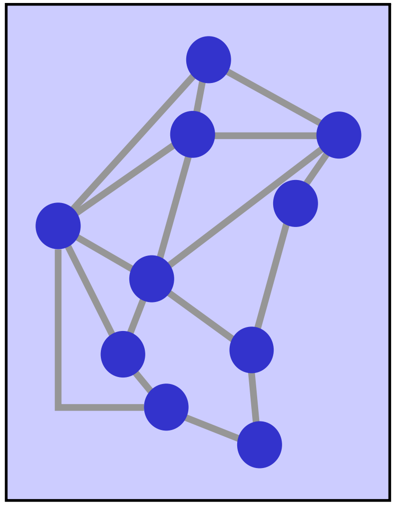
Internet Routing & Exercises

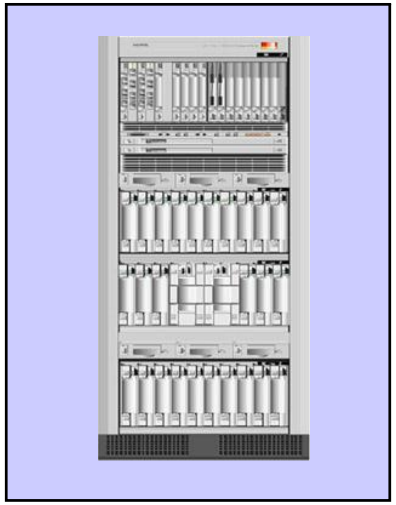
1. **Internet and Unsplittable Shortest Path Routing**
2. **The Inverse Shortest Paths Problem**
   - Exercise: Find compatible routing weights for given paths.
   - LP Models and an O(V)-approximation algorithm for ISP
3. **Shortest Path Systems**
   - Bellman property and extensions
   - General representation of SPS as …
4. **Path-based ILP model for unsplittable shortest path routing**
   - Exercise: Model as arc-flow based ILP.

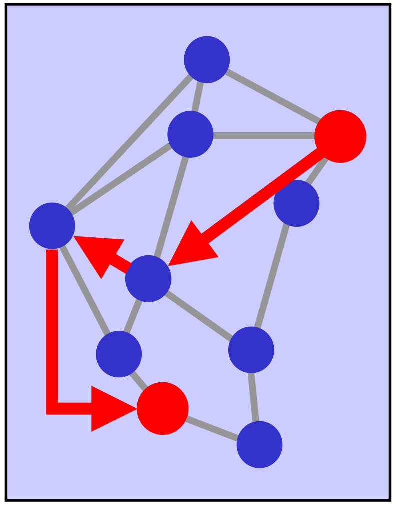
Application: Optimization of B-WiN, G-WiN, X-WiN

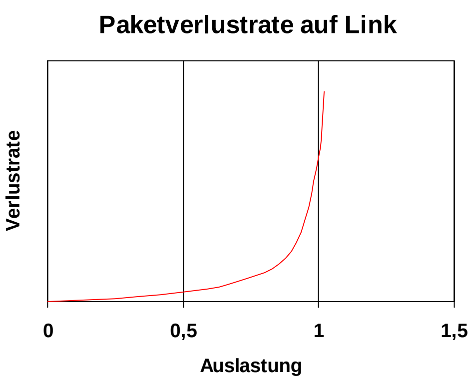
(Gigabit-Wissenschaftsnetz = Internet2 for German Universities)

IP-Network design problems









Given

- potential links
- possible link capacities and node hardware components
- end-to-end traffic demands

Decisions

- network topology
- link capacities and node hardware
- OSPF routing (weights)

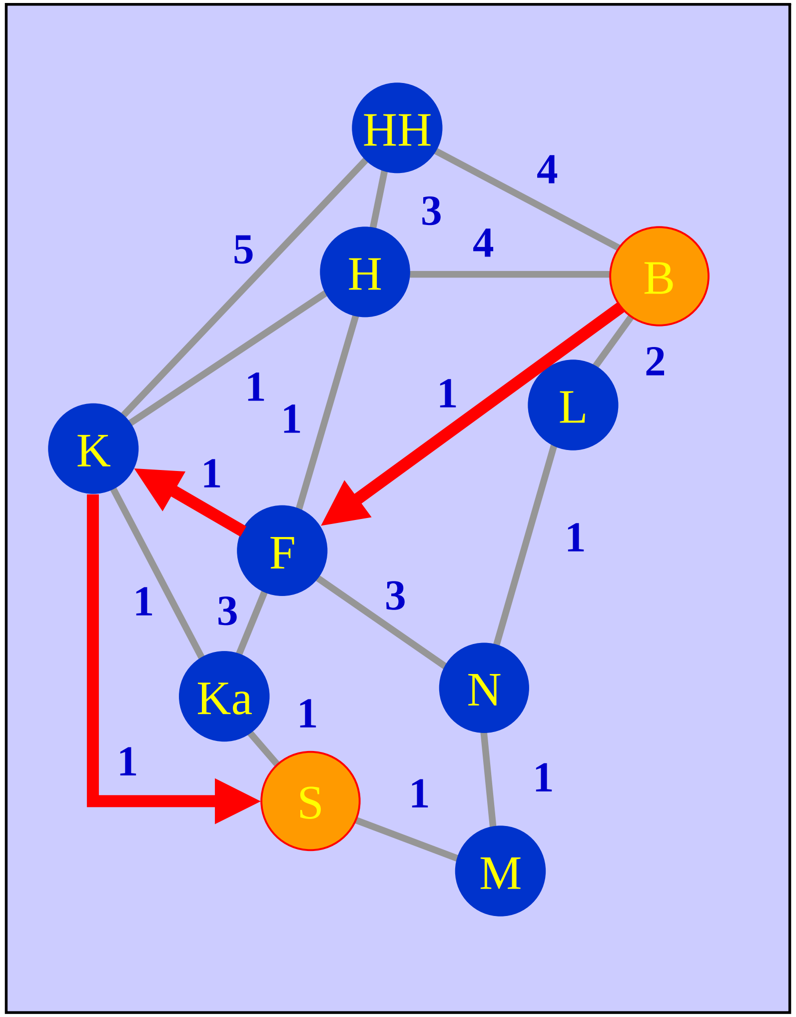
Objective

a) min link and node hardware cost
b) min maximum link load

Constraints

- hardware 'fits together'
- OSPF routing
- sufficient link capacities
- survivability

Shortest Path Routing



(1) **Set routing weights**

(Network administrator)

(2) **Compute shortest paths**

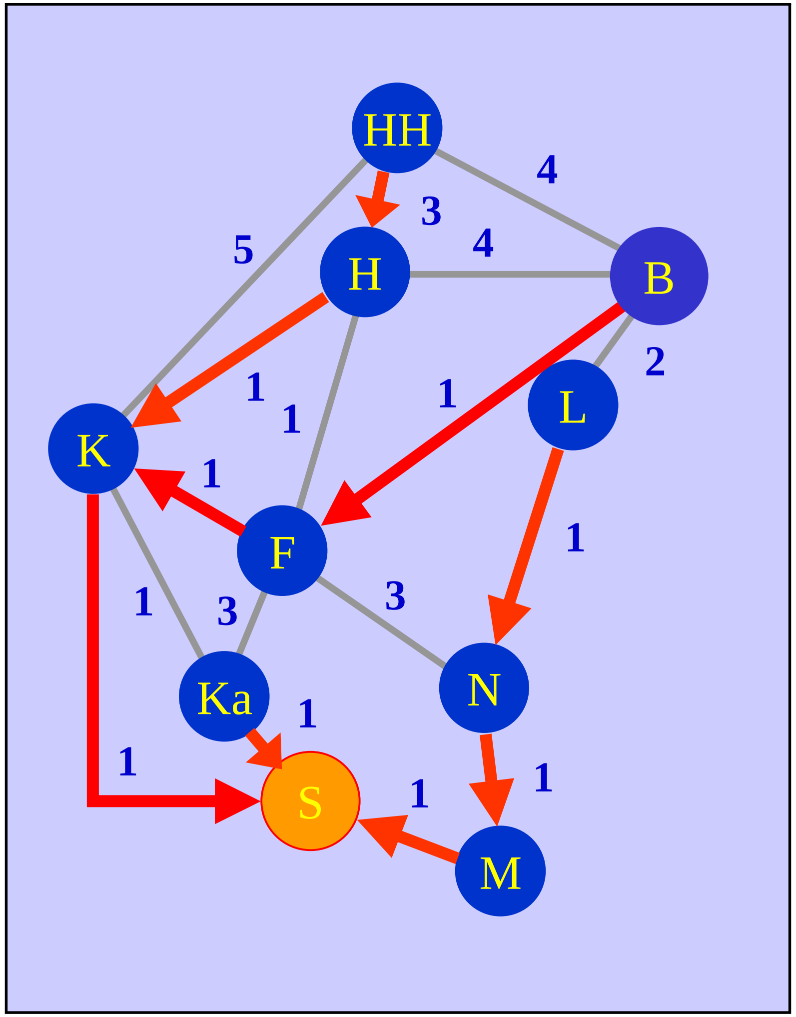
(Autonomously by routers)

(3) **Send data packets on these paths**

(Local forwarding table lookups)

Routing paths can be controlled only by changing the routing weights. (And only jointly for all paths!!!)

Shortest Path Routing



(1) **Set routing weights**
(Network administrator)

(2) **Compute shortest paths**
(Autonomously by routers)

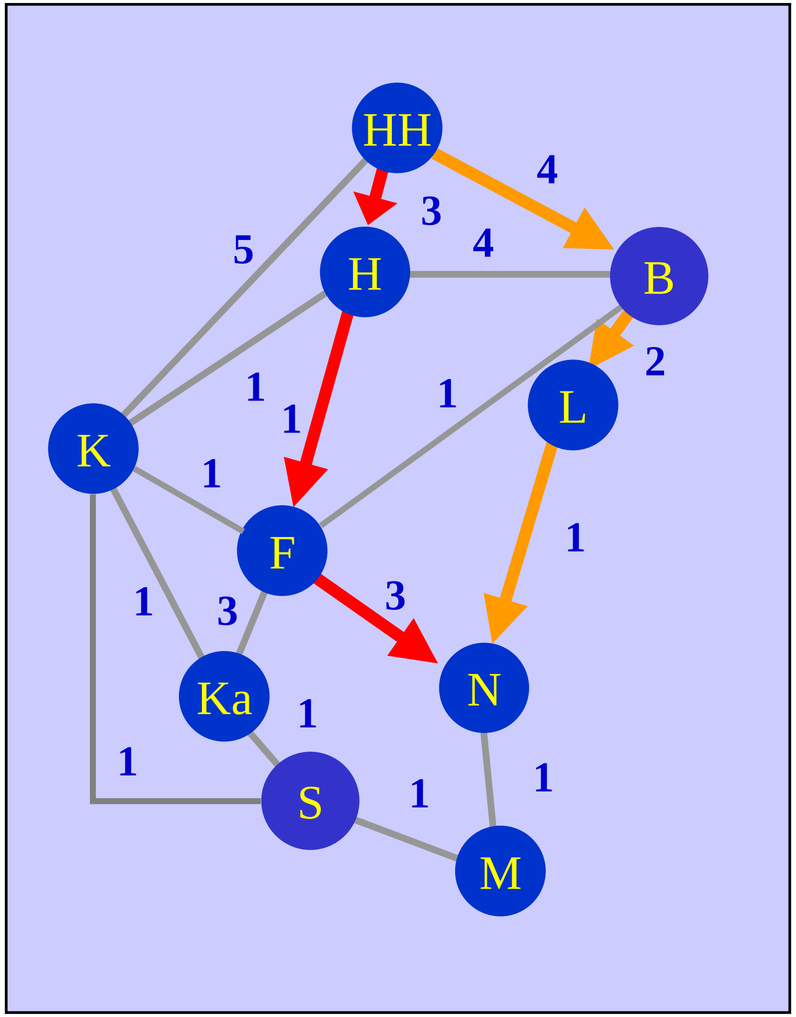
(3) **Send data packets on these paths**
(Local forwarding table lookups)

Variants:

- Distance Vector (distributed Bellman) vs. Link State (Dijkstra)
- **Single path** vs. Multi-path

Unsplittable (single) shortest path routing:
Paths to each destination form **sink-tree**

Shortest Path Routing



(1) **Set routing weights**
(Network administrator)

(2) **Compute shortest paths**
(Autonomously by routers)

(3) **Send data packets on these paths**
(Local forwarding table lookups)

Variants:

- Distance Vector vs. Link State
  (distributed Bellman) (Dijkstra)
- **Single path** vs. Multi-path

Unsplittable (single) shortest path routing:

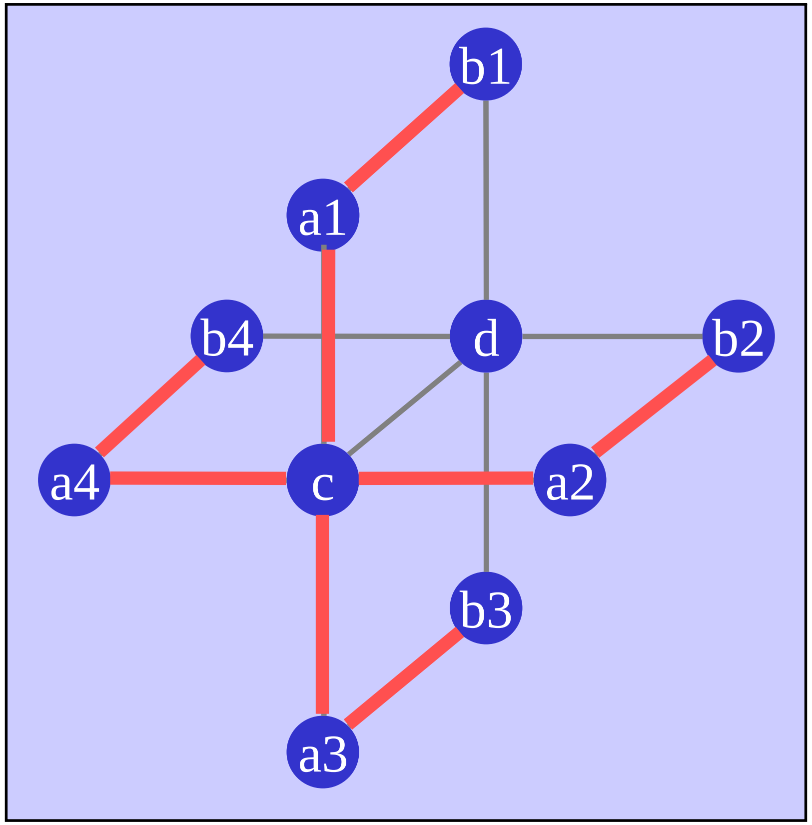
Problem: Routing is not well-defined if shortest paths are ambiguous!

Why are ambiguous shortest paths a problem?

Example: $\lambda_e = 1$ **for all** $e \in E$

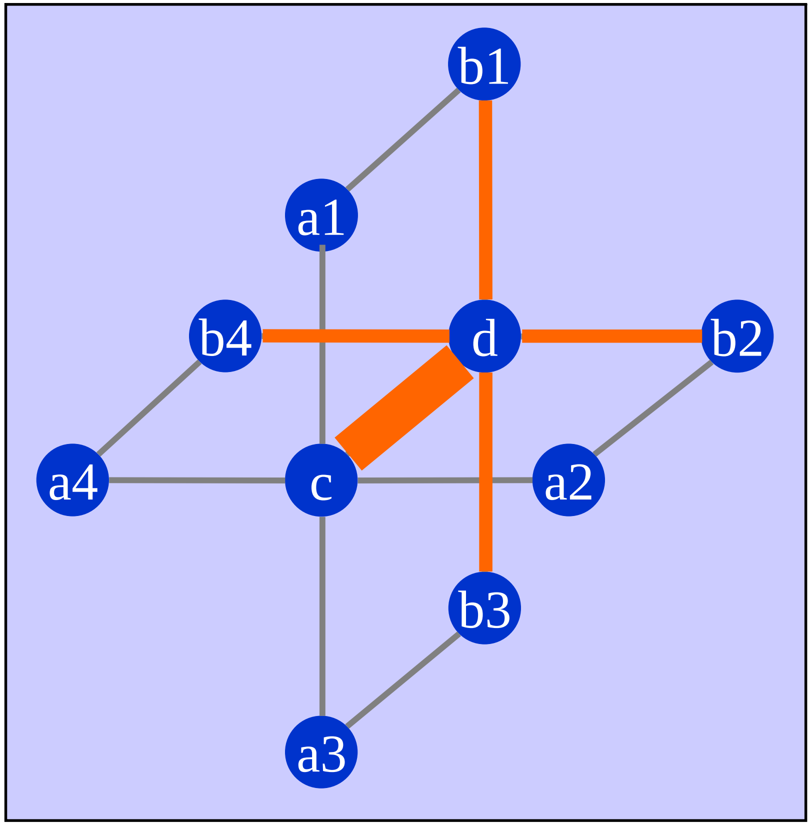
$d_{(b_i,c)} = 1$ for $i = 1,\ldots,4$

Unsplittable shortest path routing 1:



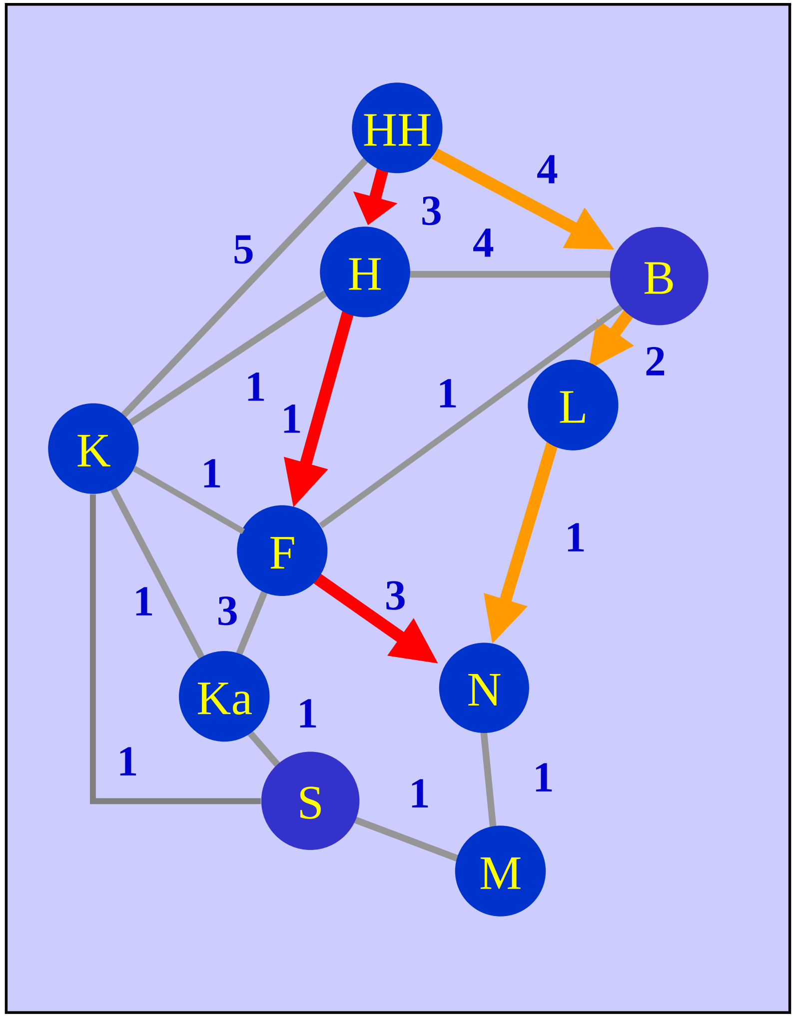
Maximum load: 1

Unsplittable shortest path routing 2:



Maximum load: 4

Shortest Path Routing



(1) **Set routing weights**
(Network administrator)

(2) **Compute shortest paths**
(Autonomously by routers)

(3) **Send data packets on these paths**
(Local forwarding table lookups)

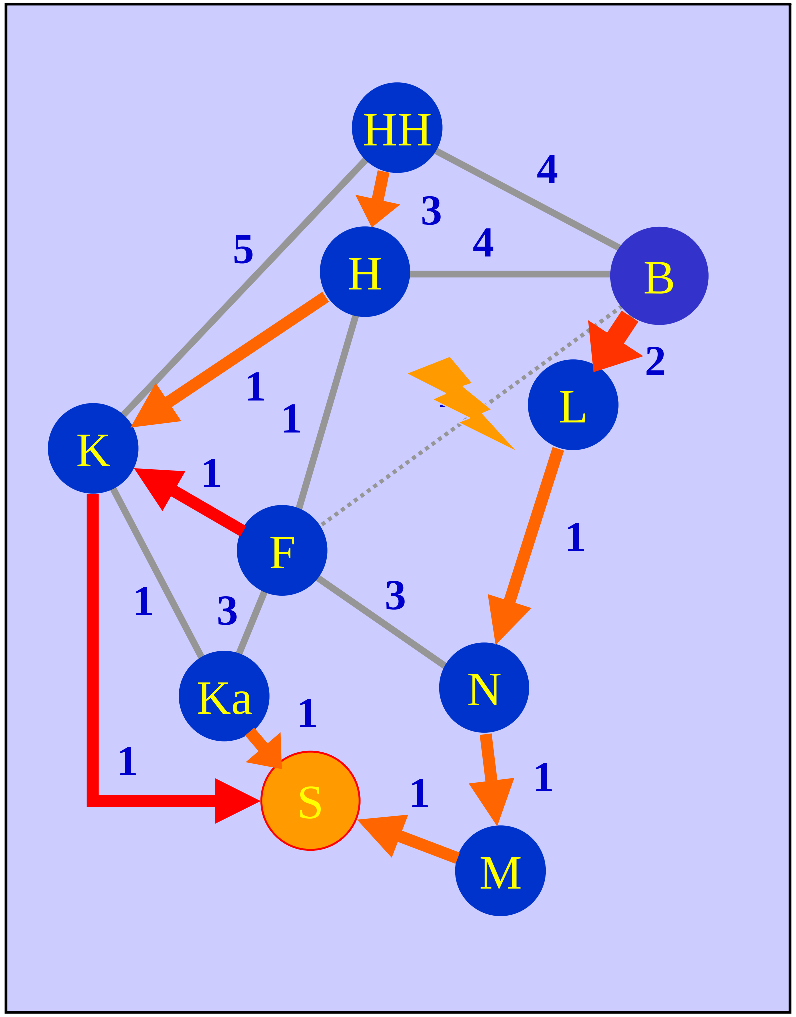
Variants:

- Distance Vector vs. Link State
  (distributed Bellman) (Dijkstra)
- **Single path** vs. Multi-path

Unsplittable (single) shortest path routing:

Routing weights must define unique shortest paths!

Shortest Path Routing – Failure Restoration



(1) **R**outing weights of failing links are set to ∞, other **weights remain unchanged**

(2) **Traffic restoration:** Recompute shortest paths in residual network

Only paths that are interrupted by failure are rerouted.

Routing weights must define unique shortest paths in normal network and in all residual networks!

Planning of Shortest Path Routing Networks

Routing consists of shortest paths for some (yet unknown) weights

Complicated interdependencies among paths of a valid routing

Weight-based approaches

Modify lengths ⇒ **Evaluate effects on routing**

- **Local Search, Genetic Algorithms, ...** [BleyGrötschelWessäly98, FarageSzentesiSzvitatovski98, FortzThorup00, EricssonResendePardalos01, BuriolResendeRibeiroThorup03, ...]
- **Lagrangian Approaches** [LinWang93, Bley03, ...]

Flow-based approaches

Optimize end-to-end flows ⇔ **Find compatible weights**

- **Integer linear programming** [Bley00, BleyKoch02, HolmbergYuan01, Prytz02, ...]

DEF: Path set $\mathcal{Q} \subseteq \mathcal{P}$ is **Shortest Path System (SPS)** if **compatible lengths** $\lambda : A \to \mathbb{R}_+$ exist (each $P \in \mathcal{Q}$ is unique shortest path).

ISP **Given:** Digraph D=(V,A) and path set Q.
Task: Find compatible lengths for Q (or prove that none exist).

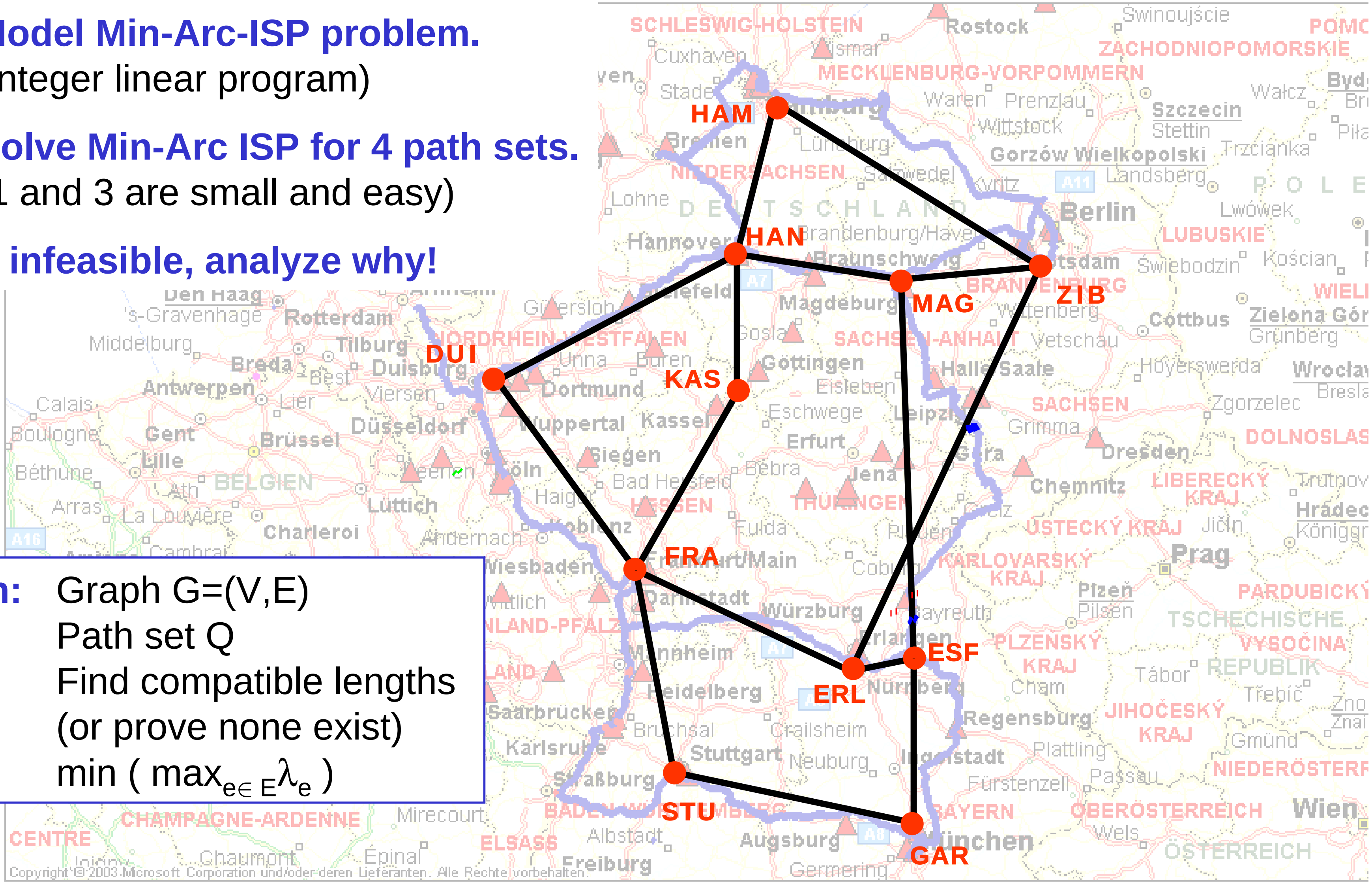
Routing protocols admit only **small integer lengths or distances**
(OSPF: $[1, \ldots, 2^{16} - 1]$, IS-IS: $[1, \ldots, 63]$, RIP: $\lambda(P) \leq 15$)

MIN-ARC-ISP

Instance: Digraph $D = (V, A)$ and path system $\mathcal{Q} \subseteq \mathcal{P}$.
Solution: Integer arc lengths $\lambda : A \to \mathbb{Z}_+$ compatible with $\mathcal{Q}$.
Objective: $\min \max_{a \in A} \lambda_a$

Exercise: Inverse Shortest Paths Problem

1. **Model Min-Arc-ISP problem.**
   (integer linear program)
2. **Solve Min-Arc ISP for 4 path sets.**
   (1 and 3 are small and easy)
3. **If infeasible, analyze why!**



Given: Graph G=(V,E)
Path set Q

Task: Find compatible lengths
(or prove none exist)

Obj: min ($\max_{e \in E} \lambda_e$)

Exercise: Inverse Shortest Paths Problem, Results

Exercise 1: PathModel.zpl (next slide)

Exercise 2 & 3:

Routing 1: λ_{max}=2

Routing 2: λ_{max}=5

Routing 3: Conflict:

P153 : DUI - HAN - MAG - ZIB

P701 : HAN - HAM – ZIB

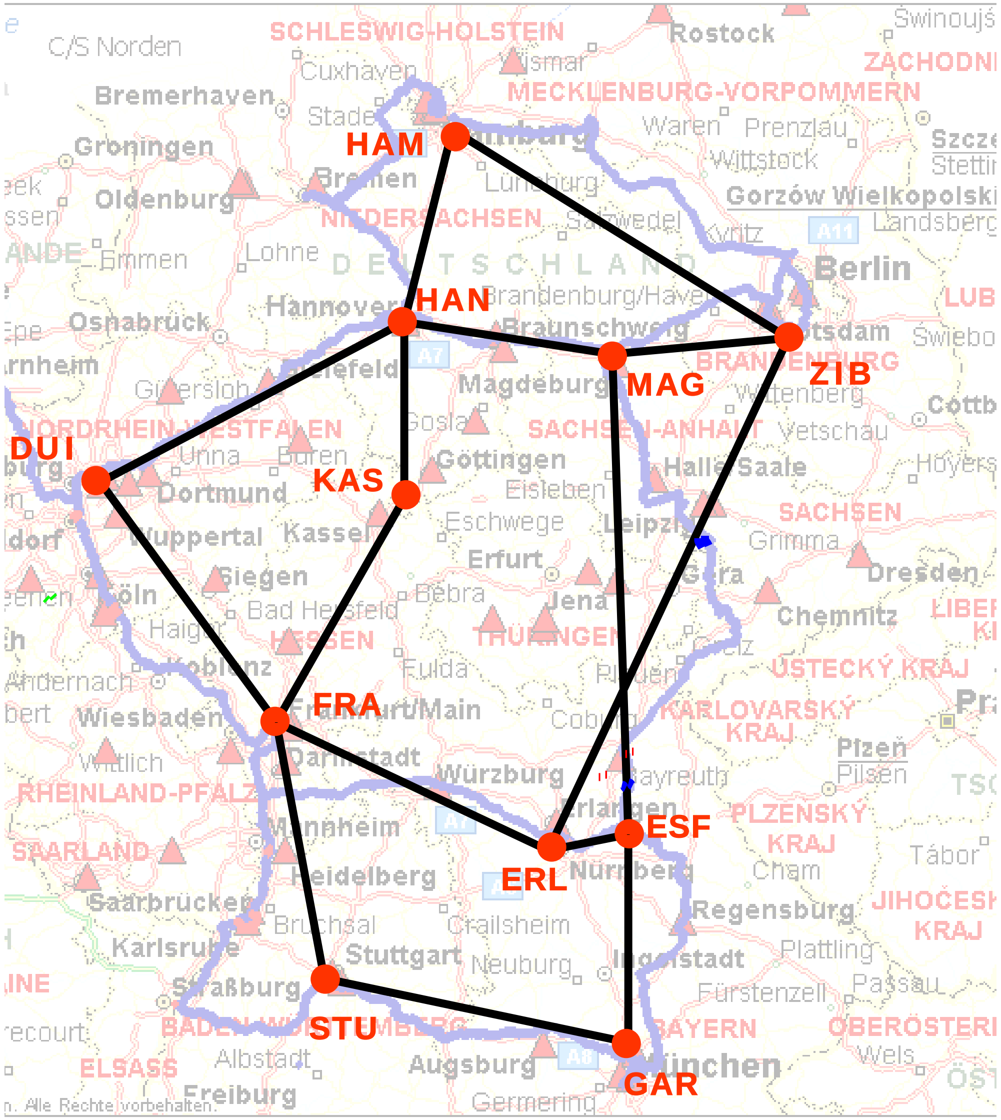
Routing 4: Conflict:

P59 : DUI - FRA - STU - GAR

P153 : DUI - HAN - MAG - ZIB

P527 : GAR - ESF - MAG - HAN - KAS

P763 : KAS - FRA - ERL – ZIB



<u>How to find Results/conflicts:</u>

> zimpl PathModel.zpl –o isp

> scip

- r isp.lp
- opt

<u>If infeasible, find unbounded dual of LP:</u>

> soplex –s0 –x isp.lp

$$\min \ \lambda_{\max}$$

$$\sum_{a \in P'} \lambda_a - \sum_{a \in P} \lambda_a \geq 1 \qquad \forall \ P \in \mathcal{Q}, \ \ P' \in \mathcal{P}(s_P, t_p) \setminus \{P\} \quad (1)$$

$$1 \leq \lambda_a \leq \lambda_{\max} \qquad \forall \ a \in A$$

$$\lambda_a \in \mathbb{Z} \qquad \forall \ a \in A$$

Model 1 is **exponentially large**, but polynomially solvable.

Exercise 1b: Devise a polynomial separation algorithm for inequalities (1).

Obs: Model 1 has an integer solution if and only if the open cone

$$\sum_{a \in P'} \lambda_a - \sum_{a \in P} \lambda_a > 0 \qquad \forall \ P \in \mathcal{Q}, \ \ P' \in \mathcal{P}(s_P, t_p) \setminus \{P\} \quad (2)$$

$$\lambda_a \geq 0 \qquad \forall \ a \in A$$

is non-empty. Any λ in this cone is a compatible metric.

Algorithm 3.1 Separate-ISP-Two-Shortest-Path

Input: $\mathcal{Q} \subseteq \mathcal{P}$ and $\lambda \in \mathbb{R}^A_+$.

Set $P^* := \emptyset$ and $\triangle^* := \infty$.

For each $P \in \mathcal{Q}$ do

Compute the shortest path P_1 and the second shortest path P_2 from s_P to t_P with respect to λ.

If $P_1 = P$ then

Set $\triangle(P) := \sum_{a \in P_2} \lambda_a - \sum_{a \in P} \lambda_a$ and $P'(P) := P_2$,

else

Set $\triangle(P) := \sum_{a \in P_1} \lambda_a - \sum_{a \in P} \lambda_a$ and $P'(P) := P_1$.

If $\triangle(P) < \triangle^*$ then

Set $P^* := P$ and $\triangle^* := \triangle(P^*)$.

If $\triangle^* < 1$ then

Return *'Inequality* (1) *is violated for* $(P^*, P'(P^*))$.*'*

else

Return *'All inequalities* (1) *are satisfied.'*

Inverse Shortest Paths Problem: Model 2

$$
\begin{aligned}
& \min\ \lambda_{\max} \\
& d_{(s,v)} + \lambda_{(v,t)} - d_{(s,t)} = 0 && \forall\ P \in \mathcal{Q},\ (v,t) \in P[s,t] \neq \emptyset \\
& d_{(s,v)} + \lambda_{(v,t)} - d_{(s,t)} \geq 1 && \forall\ P \in \mathcal{Q},\ (v,t) \notin P[s,t] \neq \emptyset \\
& d_{(s,t)} \geq 0 && \forall\ s,t \in V \\
& 1 \leq \lambda_a \leq \lambda_{\max} && \forall\ a \in A \\
& \lambda_a \in \mathbb{Z}_+
\end{aligned}
$$

Model 2 is **polynomially large**: $O(|V|^2)$ variables and $O(|Q||V|^3)$ constraints.

Algorithm ISP-Rounding

1. Solve the following linear program:

$$\min \ \lambda_{\text{max}}$$

$$\sum_{a \in P'} \lambda_a - \sum_{a \in P} \lambda_a \geq |V|/2 \quad \forall\ P \in \mathcal{Q},\ \ P' \in \mathcal{P}(s_P, t_p) \setminus \{P\} \quad (3)$$

$$0.5 \leq \lambda_a \leq \lambda_{\text{max}} \quad \forall\ a \in A$$

2. Round optimal solution λ^*

Thm: ISP-Rounding is a |V|/2-approximation algorithm for Min-Arc-ISP.

Proof: $\lambda^*_{max} \leq |V|/2$ Opt(LP-Relaxation of Model 1) $\leq |V|/2$ Opt(Model 1)
$|P' \cup P| \leq |V|$ for all pairs P´,P in inequality (3)
At least one arc $a \in P'$ appears in each inequality (3)
Hence, $\lceil \lambda^* \rceil$ satisfies all inequalities (2).

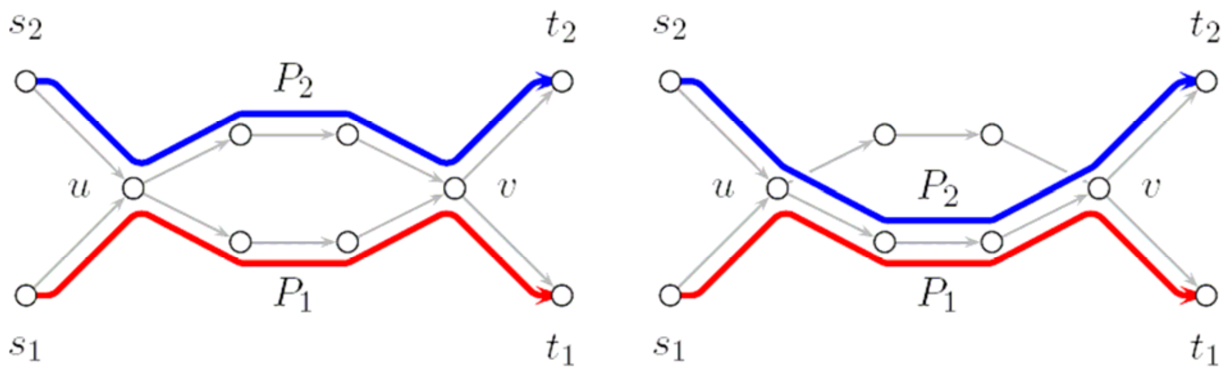
Further Results for Inverse Shortest Paths

Thm [BenAmeurGourdin00]: Min-Arc-ISP and Min-Path-ISP are approximable within a factor of min($|V|/2$, $\max_{P\in Q}|P|$).

Thm [B'04]: It is NP-hard to approximate Min-Arc-ISP within a factor of $9/8-\varepsilon$, for any $\varepsilon>0$.

Thm [B'04]: It is NP-hard to approximate Min-Path-ISP within a factor of $8/7-\varepsilon$, for any $\varepsilon>0$.

Bellman property



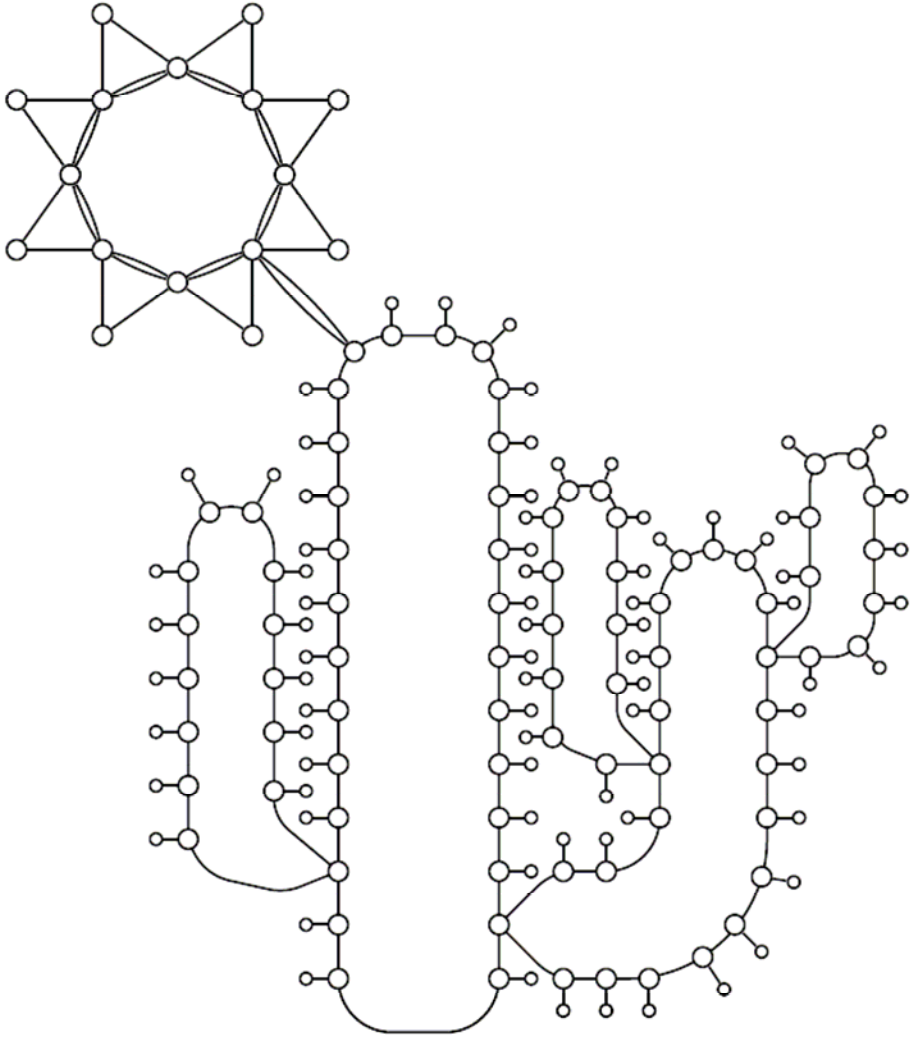
(a) Paths P_1 and P_2 are conflicting. (b) Paths P_1 and P_2 have B-property.

Def: P_1 and P_2 have the B-property if $P_1[u,v]=P_2[u,v]$ for all u,v with $P_1[u,v] \neq \emptyset$ and $P_2[u,v] \neq \emptyset$. Otherwise P_1 and P_2 conflict.

Obs: Any SPS has B-property.

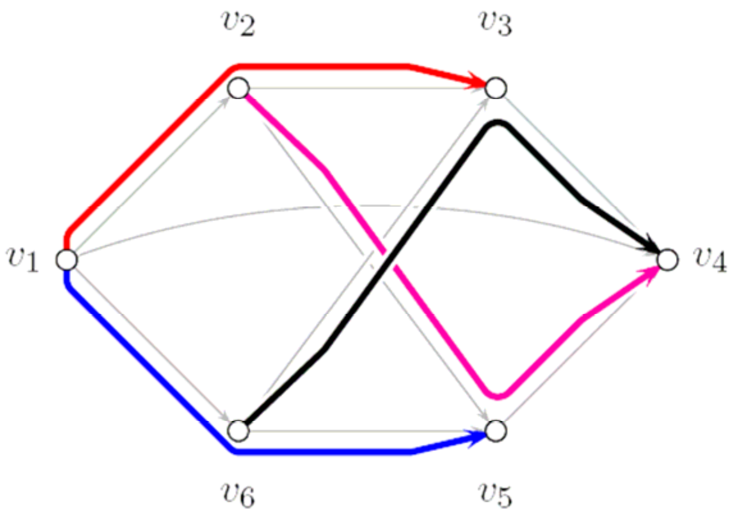
Thm: Any path set $|Q| \leq 3$ with B-property is an SPS.

Obs [BenAmeur00]: In undirected cactus graphs, any path set with the B-property is an SPS.



Bellman property

Obs: There are non-SPS path sets with B-property.

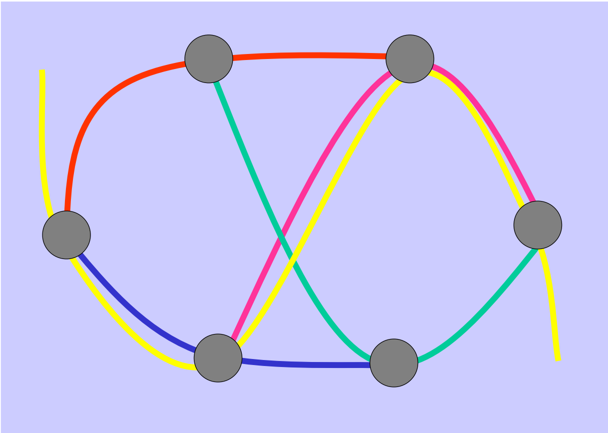


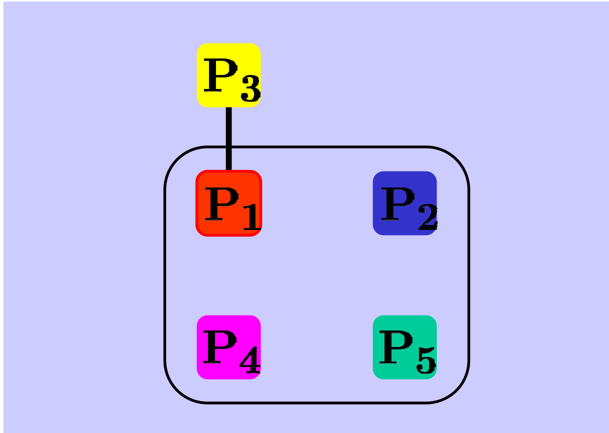
$P_1 = (v_1, v_2, v_3)$
$P_2 = (v_1, v_6, v_5)$
$P_3 = (v_2, v_5, v_4)$
$P_4 = (v_6, v_3, v_4)$

In the subgraph $D-(1,4)$, any (v_1, v_4)-path P' conflicts with some $P \in \{P_1, P_2, P_3, P_4\}$.

OBS: Shortest Path Systems form an independence system $\mathcal{I}_{SPS} \subseteq 2^{\mathcal{P}}$, but not a matroid.

Representation: weakly stable sets in conflict hypergraph $(\mathcal{P}, \mathcal{C}_{SPS})$





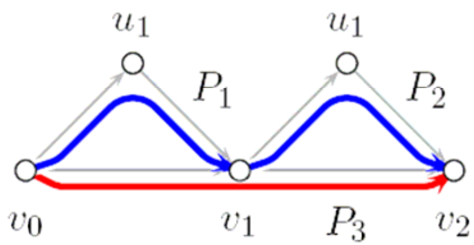
Maximal SPS	= bases in indep. system	= maximal weakly stable sets
Minimal Non-SPS	= circuit in indep. system	= conflict hyperedges
Conflicting paths	= rank 1 circuits	= simple conflict edges

Shortest Path Systems

OBS: Shortest Path Systems form an independendence system $\mathcal{I}_{SPS} \subseteq 2^{\mathcal{P}}$, but not a matroid.

Example: Why is $\mathcal{I}_{SPS} \subseteq 2^{\mathcal{P}}$ not a matroid?

$\{P_1,P_2\}$ is an SPS and $\{P_3\}$ is an SPS.
$\{P_1,P_3\}$ and $\{P_2,P_3\}$ are no SPSs.



$$P_1 = (v_0, u_1, v_1)$$
$$P_2 = (v_1, u_2, v_2)$$
$$P_3 = (v_0, v_1, v_2)$$

Obs: The rank quotient of $\mathcal{I}_{SPS}$ may become arbitrarily large.

Theorem: One can decide polynomially whether $\mathcal{Q} \in \mathcal{I}_{SPS}$ or not.

Corollary: Given a non-SPS $\mathcal{Q} \notin \mathcal{I}_{SPS}$, one can find in polynomial time an irreducible non-SPS $\mathcal{P} \in \mathcal{C}_{SPS}$ with $\mathcal{P} \subseteq \mathcal{Q}$.

Algorithm: Greedily remove paths from Q and check if rest is SPS.

Theorem [B'04]: Finding the minimum cardinality or minimum weight irreducible non-SPS $\mathcal{P} \subseteq \mathcal{Q}$ for $\mathcal{Q} \notin \mathcal{I}_{SPS}$ is NP-hard.

Theorem [B'04]: Finding the maximum cardinality or maximum weight SPS $\mathcal{P} \subseteq \mathcal{Q}$ for some $\mathcal{Q} \notin \mathcal{I}_{SPS}$ is NP-hard.

Corollary: Computing the rank of an arbitrary path set is NP-hard.

ILP models for capacitated unsplittable shortest path routing

Given: Digraph D=(V,A) with capacities c_a
Commodity set $K \subset V^2$ with demands $d_{(s,t)}$
Task: Find USPR such that the flows do not exceed the capacities.

$$\sum_{P \in \mathcal{P}(s,t)} x_P = 1 \qquad \forall\ (s,t) \in K \qquad (1)$$

$$x_P \geq 0 \qquad \forall\ P \in \mathcal{P} \qquad (2)$$

$$x_P \in \mathbb{Z} \qquad \forall\ P \in \mathcal{P} \qquad (3)$$

$$\sum_{P:\ a \in P} d_{(s_P,t_P)} x_P \leq c_a \qquad \forall\ a \in A \qquad (4)$$

$$\sum_{P \in \mathcal{Q}} x_P \leq |\mathcal{Q}| - 1 \qquad \forall\ \mathcal{Q} \in \mathcal{C}_{SPS} \qquad (5)$$

(1)-(3): Choose one path for each commodity.
(4): Flows do not exceed the capacities.
(5): The paths must form an SPS (i.e., there is a compatible metric).

ILP models for capacitated unsplittable shortest path routing

Thm: (1)-(5) is a correct model for CapUSPR.

Proof: (1)-(4) is a correct model for capacitated unsplittable flow.
(5) ensures that no integer solution \`contains´ an (irreducible) non-SPS.

Model (1)-(5) contains **exponentially many variables** and **exponentially many constraints**.

Thm: There are instances, where the optimal solution of the linear programming relaxation of (1)-(5) has exponentially many active path variables x_P.

ILP models for capacitated unsplittable shortest path routing

Thm: Separation problem for inequalities (5) is NP-hard for $x \in [0,1]^P$.

Proof: Equivalent to finding a minimum weight non-SPS $\mathcal{Q}$.

Thm: Separation problem for inequalities (5) is polynomial for $x \in \{0,1\}^P$.

Proof: For $x \in \{0,1\}^P$, inequality (5) is violated for all irreducible non-SPS $\mathcal{Q} \subseteq \{P : x_p = 1\}$, and only for those.
Greedily remove paths from $\{P : x_p = 1\}$ and check whether the rest is an SPS or not.

We can at least cut-off infeasible binary vectors $x \in \{0,1\}^P$ efficiently in a Branch-and-Cut Framework based on formulation (1)-(5).

Model (1)-(5) is intersection of
- Capacitated unsplittable flow polytope UFP and
- $IND(\mathcal{I}_{SPS}) = STAB((\mathcal{P}, \mathcal{C}_{SPS})) = conv\{\chi^{\mathcal{Q}} : \mathcal{Q}$ is SPS$\}$

Cor: Any valid inequality for UFP and $IND(\mathcal{I}_{SPS})$ is valid for (1)-(5), too.

Rank inequalities: $$\sum_{P \in \mathcal{Q}} x_P \leq r(Q) \quad \forall\ \mathcal{Q} \subseteq \mathcal{P}$$

Contains **clique** and odd hole **inequalities in the conflict (hyper)graph**.

Thm: Separation of rank inequalities is NP-hard. (Even computing the rhs of a given set is NP-hard!)

Thm: Gap between $IND(\mathcal{I}_{SPS})$ and its linear relaxation with rank inequalities $RK(\mathcal{I}_{SPS})$ may become arbitrarily large.

Joint inequalities induced by shortest path routing + capacities:
Induced cover inequalities

Every arc capacity defines a knapsack with precedence constraints:

$$\sum_{P:\ a \in P} d_{(s_P, t_P)} x_P \leq c_a$$

$$x'_P \leq x_P \qquad \forall\ P \text{ is subpath of } P'$$

$$x_P \in \{0, 1\} \quad \forall\ P : a \in P$$

Induced cover is a set $\mathcal{Q} \subseteq \{P : a \in P\} \quad : \quad \sum_{P:P \text{ is subpath of } P' \in \mathcal{Q}} d_{(s_P, t_P)} > c_a$

Induced cover inequality: $\sum_{P' \in \mathcal{Q}} x_{P'} \leq |\mathcal{Q}| - 1$

Precedence graph has bounded tree width ⇒ separable via **dyn. prog.**

Solution approach

Mixed-integer programming model

Variables

(Link capacities etc.)

Path or Arc-flow variables

Constraints

(Admissible hardware configuration)

Capacity constraints

Flow conservation and integrality

Shortest path routing (easy)

Shortest path routing (hard)

Algorithms

Network design and end-to-end routing

- Cutting plane algorithm
- Branch & Cut (& Price)
- Heuristics

Separation of (5)

Compatible routing weights

Linear programming

Traffic Engineering: Results

Task: Reduce maximum link load by optimizing the routing weights.