Resilient Capacity-Aware Routing

Stefan Schmid[1], Nicolas Schnepf[2] and Jiri Srba[2]

[1] University of Vienna, Austria
[2] Aalborg University, Denmark

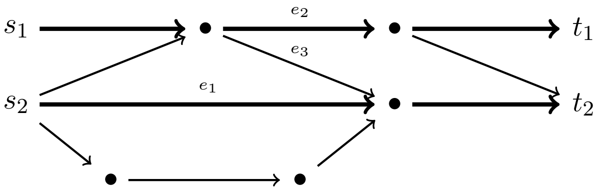
TACAS 2021

Routing in Computer Networks

- Increasing demands on available and dependable networks.
- Networks must quickly adapt their routing in case of link failures.
- Connectivity of a network under failures is only a necessary condition.
- Resilient networks must also respect capacity constraints on links.

Routing in Computer Networks

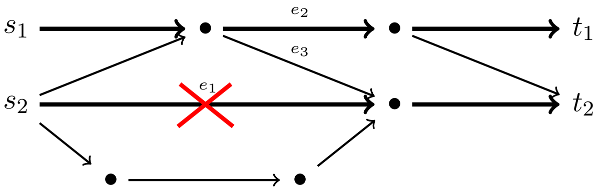
- Increasing demands on available and dependable networks.
- Networks must quickly adapt their routing in case of link failures.
- Connectivity of a network under failures is only a necessary condition.
- Resilient networks must also respect capacity constraints on links.



- Two flows: s_1 to t_1 and s_2 to t_2, both with demand 1.
- All links have capacity 1 and weight 1.
- Shortest paths routing—Equal-Cost-Multipath Protocol (ECMP).

Routing in Computer Networks

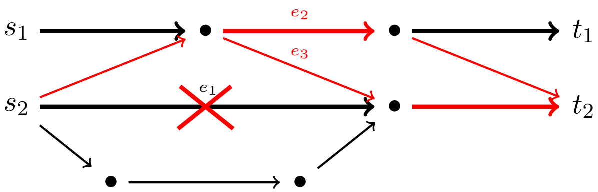
- Increasing demands on available and dependable networks.
- Networks must quickly adapt their routing in case of link failures.
- Connectivity of a network under failures is only a necessary condition.
- Resilient networks must also respect capacity constraints on links.



- Two flows: s_1 to t_1 and s_2 to t_2, both with demand 1.
- All links have capacity 1 and weight 1.
- Shortest paths routing—Equal-Cost-Multipath Protocol (ECMP).

Routing in Computer Networks

- Increasing demands on available and dependable networks.
- Networks must quickly adapt their routing in case of link failures.
- Connectivity of a network under failures is only a necessary condition.
- Resilient networks must also respect capacity constraints on links.



- Two flows: s_1 to t_1 and s_2 to t_2, both with demand 1.
- All links have capacity 1 and weight 1.
- Shortest paths routing—Equal-Cost-Multipath Protocol (ECMP).

Studies Problem and Results

Question

Can traffic (set of flows) in a network with up to k failed links be forwarded along shortest paths without violating the capacity constraints?

Studies Problem and Results

Question

Can traffic (set of flows) in a network with up to k failed links be forwarded along shortest paths without violating the capacity constraints?

Our Contribution

- We introduce the pessimistic (P) and optimistic (O) routing policy.
- Consider both splittable (S) and nonsplittable (N) flows.
- A complete complexity characterization of deciding the existence of a routing policy without ($k = 0$) and with ($k > 0$) link failures.
- Develop an efficient strategic search algorithm and demonstrate its performance on a large number of datacenter and ISP networks.

Network with Capacities and Demands

Network with Capacities and Demands (NCD) is $N = (V, C, D)$ where

- V is a finite set of nodes (switches and routers),
- $C : V \times V \mapsto \mathbb{N}^0$ is the *capacity function*, and
- $D : V \times V \mapsto \mathbb{N}^0$ is the end-to-end *flow demand*.

- The routing of flow demands is the shortest paths (ECMP).
- Network operator defines the shortest path by *weight assignment* to links $W : V \times V \mapsto \mathbb{N} \cup \{\infty\}$.

Flow Assignment

A *flow assignment* f for $F \subseteq V \times V$ of failed links is a family of functions

$$f^F_{s,t}: \ SPaths^F(s,t) \mapsto [0,1]$$

for all $s, t \in V$ such that

$$\sum_{\pi \in SPaths^F(s,t)} f^F_{s,t}(\pi) = 1 \ .$$

- f is *nonsplittable* if $f^F_{s,t}(\pi) \in \{0,1\}$, otherwise it is *splittable*.

Flow Assignment

A *flow assignment* f for $F \subseteq V \times V$ of failed links is a family of functions

$$f^F_{s,t} : \; SPaths^F(s,t) \mapsto [0,1]$$

for all $s, t \in V$ such that

$$\sum_{\pi \in SPaths^F(s,t)} f^F_{s,t}(\pi) = 1 \; .$$

- f is *nonsplittable* if $f^F_{s,t}(\pi) \in \{0,1\}$, otherwise it is *splittable*.

A flow assignment f is *feasible* if for every link (v, v'):

$$\sum_{\substack{s,t \in V \\ \pi \in SPaths^F(s,t) \\ (v,v') \in \pi}} f^F_{s,t}(\pi) \cdot D(s,t) \leq C(v,v') \; .$$

Problems Definition

Pessimistic Splittable/Nonsplittable (PS/PN)

Given a nonnegative integer k, is it the case that for *every* set F of failed links where $|F| \leq k$, the network remains connected and *every* splittable/nonsplittable flow assignment on N is feasible?

Problems Definition

Pessimistic Splittable/Nonsplittable (PS/PN)

Given a nonnegative integer k, is it the case that for *every* set F of failed links where $|F| \leq k$, the network remains connected and *every* splittable/nonsplittable flow assignment on N is feasible?

Optimistic Splittable/Nonsplittable (OS/ON)

Given a nonnegative integer k, *is there* a feasible splittable/nonsplittable flow assignment for *every* set of failed links F where $|F| \leq k$?

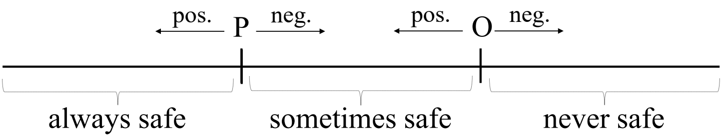
Problems Definition

Pessimistic Splittable/Nonsplittable (PS/PN)

Given a nonnegative integer k, is it the case that for *every* set F of failed links where $|F| \leq k$, the network remains connected and *every* splittable/nonsplittable flow assignment on N is feasible?

Optimistic Splittable/Nonsplittable (OS/ON)

Given a nonnegative integer k, *is there* a feasible splittable/nonsplittable flow assignment for *every* set of failed links F where $|F| \leq k$?



Complexity Results

$k = 0$	**Pessimistic**	**Optimistic**
Splittable	NL-complete	P-complete
Nonsplitable	NL-complete	NP-complete

Complexity Results

$k = 0$	**Pessimistic**	**Optimistic**
Splittable	NL-complete	P-complete
Nonsplitable	NL-complete	NP-complete

$k > 0$	**Pessimistic**	**Optimistic**
Splittable	co-NP-complete	co-NP-complete
Nonsplitable	co-NP-complete	Π_2^P-complete

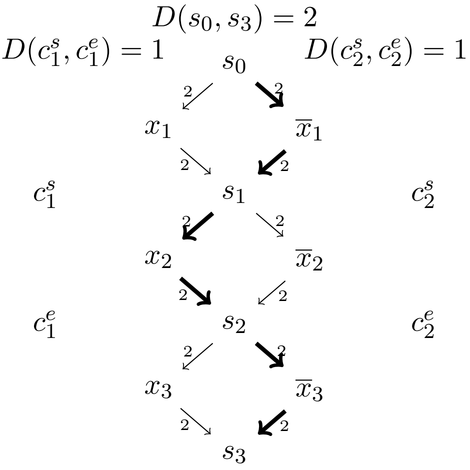
Π_2^P-Containment for ON-Problem

We want to check whether **for all** failure scenarios, **exists** a feasible nonsplittable flow assignment?

Existence of flow assignment for a fixed failure scenario by ILP.

$$
\begin{aligned}
0 \le x^{s,t}(v,v') \le 1 \quad & \text{for } s,t,v,v' \in V \\
\sum_{v \in V} x^{s,t}(s,v) \cdot spg^{s,t}(s,v) = 1 \quad & \text{for } s,t \in V \\
\sum_{v \in V} x^{s,t}(v,t) \cdot spg^{s,t}(v,t) = 1 \quad & \text{for } s,t \in V \\
\sum_{v' \in V} x^{s,t}(v',v) \cdot spg^{s,t}(v',v) = & \\
\sum_{v' \in V} x^{s,t}(v,v') \cdot spg^{s,t}(v,v') \quad & \text{for } s,t,v \in V, v \notin \{s,t\} \\
\sum_{s,t \in V} x^{s,t}(v,v') \cdot spg^{s,t}(v,v') \cdot D(s,t) \le C(v,v') \quad & \text{for } v,v' \in V
\end{aligned}
$$

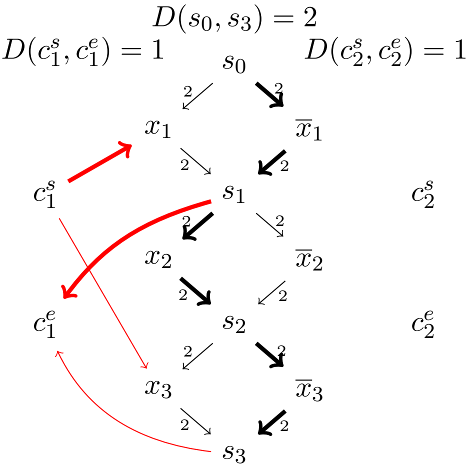
Π_2^P-Hardness for ON-Problem



$\exists x_1, x_2, x_3.$

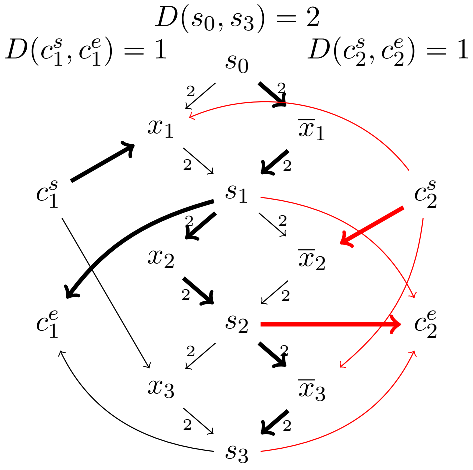
$(x_1 \vee x_3) \wedge (x_1 \vee \overline{x}_2 \vee \overline{x}_3)$

Π_2^P-Hardness for ON-Problem



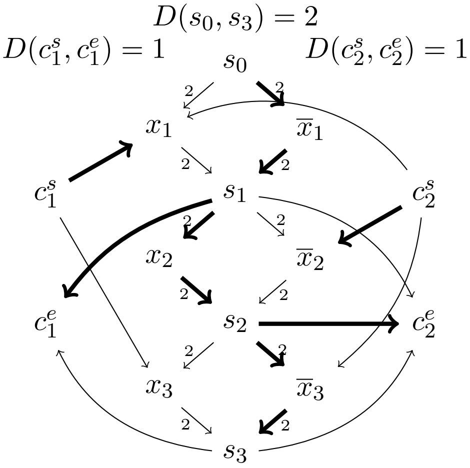
$\exists x_1, x_2, x_3.$
$(x_1 \vee x_3) \wedge (x_1 \vee \overline{x}_2 \vee \overline{x}_3)$

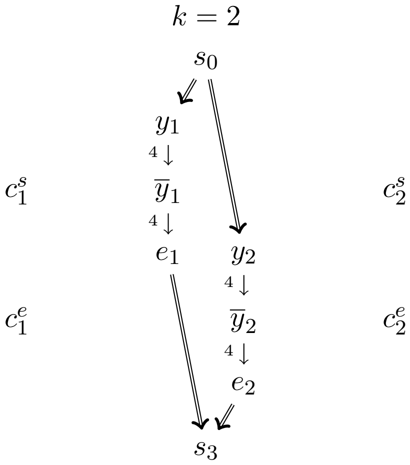
Π_2^P-Hardness for ON-Problem



$\exists x_1, x_2, x_3.$
$(x_1 \vee x_3) \wedge (x_1 \vee \overline{x}_2 \vee \overline{x}_3)$

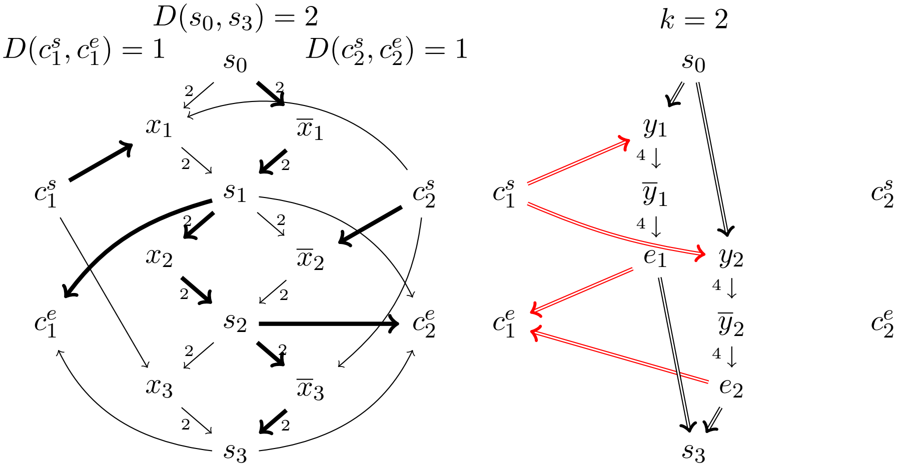
Π_2^P-Hardness for ON-Problem





$\exists x_1, x_2, x_3.$
$(x_1 \vee x_3) \wedge (x_1 \vee \overline{x}_2 \vee \overline{x}_3)$

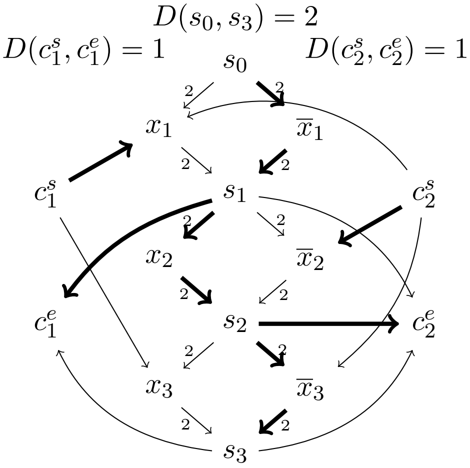
$\forall y_1, y_2.$ $\exists x_1, x_2, x_3.$
$(x_1 \vee x_3 \vee y_1 \vee \overline{y}_1 \vee \overline{y}_2) \wedge$
$(x_1 \vee \overline{x}_2 \vee \overline{x}_3 \vee y_2)$

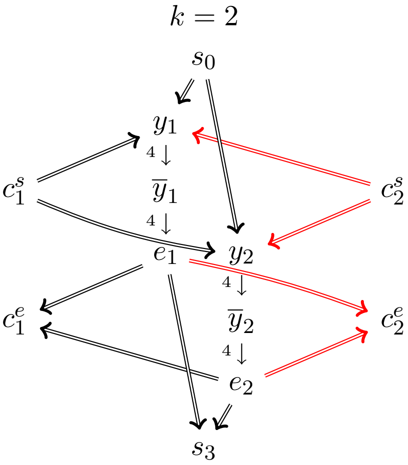


$\exists x_1, x_2, x_3.$
$(x_1 \vee x_3) \wedge (x_1 \vee \overline{x}_2 \vee \overline{x}_3)$

$\forall y_1, y_2.$ $\exists x_1, x_2, x_3.$
$(x_1 \vee x_3 \vee y_1 \vee \overline{y}_1 \vee \overline{y}_2) \wedge$
$(x_1 \vee \overline{x}_2 \vee \overline{x}_3 \vee y_2)$

Π_2^P-Hardness for ON-Problem

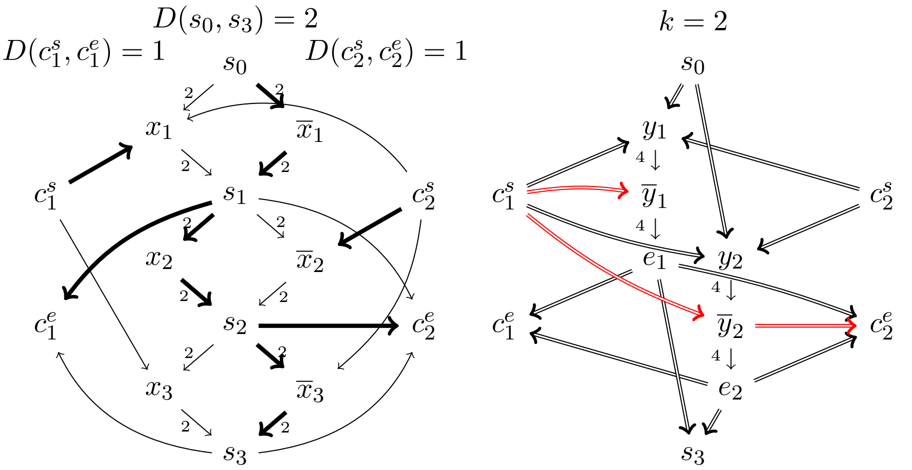




$\exists x_1, x_2, x_3.$
$(x_1 \lor x_3) \land (x_1 \lor \overline{x}_2 \lor \overline{x}_3)$

$\forall y_1, y_2.\ \exists x_1, x_2, x_3.$
$(x_1 \lor x_3 \lor y_1 \lor \overline{y}_1 \lor \overline{y}_2) \land$
$(x_1 \lor \overline{x}_2 \lor \overline{x}_3 \lor y_2)$

Π_2^P-Hardness for ON-Problem



$\exists x_1, x_2, x_3.$
$(x_1 \vee x_3) \wedge (x_1 \vee \overline{x}_2 \vee \overline{x}_3)$

$\forall y_1, y_2.\ \exists x_1, x_2, x_3.$
$(x_1 \vee x_3 \vee y_1 \vee \overline{y}_1 \vee \overline{y}_2) \wedge$
$(x_1 \vee \overline{x}_2 \vee \overline{x}_3 \vee y_2)$

Solving the Capacity Problems by Brute-Force

Input: NCD $N = (V, C, D)$ with weigth assignment W, a number $k \geq 0$ and type of the capacity problem $\tau \in \{\mathsf{PS}, \mathsf{PN}, \mathsf{ON}, \mathsf{OS}\}$
Output: *true* if the answer to the τ-problem is positive, else *false*

for all $F \subseteq V \times V$ s.t. $|F| \leq k$ **do**
  construct network N^F by removing from N the failed links F
  switch τ **do**
    case OS: use LP to solve N^F for $k = 0$
    case ON: use ILP to solve N^F for $k = 0$
    case PS/PN: use NL-algorithm to solve N^F for $k = 0$
  if the answer to the τ-problem on N^F is negative **then return** *false*
endfor
return *true*

Strategic Search Algorithm

Main idea of strategic search algorithm: reduce the number of explored failure scenarios by skipping the "uninteresting" ones.

Precedence relation on failure scenarios

We write $F \prec F'$ if for all $s, t \in V$:

- $F \subseteq F'$ and
- $SPaths^{F}(s,t) \cap SPaths^{F'}(s,t) \neq \emptyset$.

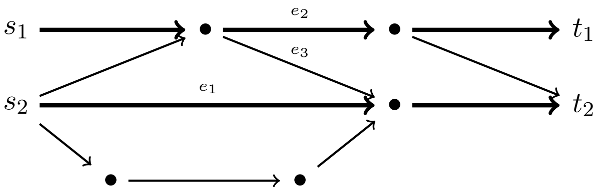
Strategic Search Algorithm

Main idea of strategic search algorithm: reduce the number of explored failure scenarios by skipping the “uninteresting” ones.

Precedence relation on failure scenarios

We write $F \prec F'$ if for all $s, t \in V$:

- $F \subseteq F'$ and
- $SPaths^{F}(s,t) \cap SPaths^{F'}(s,t) \neq \emptyset$.



$\{e_1\} \prec \{e_1, e_3\}$ however $\{e_1\} \not\prec \{e_1, e_2\}$ and $\{e_1, e_2\} \not\prec \{e_1, e_2, e_3\}$

Observations Useful for Strategic Search

Lemma (pessimistic cases)

Let $F \prec F'$. A positive answer to the PS/PN problem for the network N^F implies a positive answer to the PS/PN problem for the network $N^{F'}$.

It is enough to explore only the minimum failure scenarios w.r.t. $\prec$.

Observations Useful for Strategic Search

Lemma (pessimistic cases)

Let $F \prec F'$. A positive answer to the PS/PN problem for the network N^F implies a positive answer to the PS/PN problem for the network $N^{F'}$.

It is enough to explore only the minimum failure scenarios w.r.t. $\prec$.

Lemma (optimistic cases)

Let $F \prec F'$. A positive answer to the OS/ON problem for the network $N^{F'}$ implies a positive answer to the OS/ON problem for the network N^F.

It is enough to explore only the maximum failure scenarios w.r.t. $\prec$.

Observations Useful for Strategic Search

Lemma (pessimistic cases)

Let $F \prec F'$. A positive answer to the PS/PN problem for the network N^F implies a positive answer to the PS/PN problem for the network $N^{F'}$.

It is enough to explore only the minimum failure scenarios w.r.t. $\prec$.

Lemma (optimistic cases)

Let $F \prec F'$. A positive answer to the OS/ON problem for the network $N^{F'}$ implies a positive answer to the OS/ON problem for the network N^F.

It is enough to explore only the maximum failure scenarios w.r.t. $\prec$.

Identification of minimum/maximum failure scenarios

To find minimum/maximum failure scenarios w.r.t. $\prec$, we find minimum s-t cuts that disconnet the source s and target t.

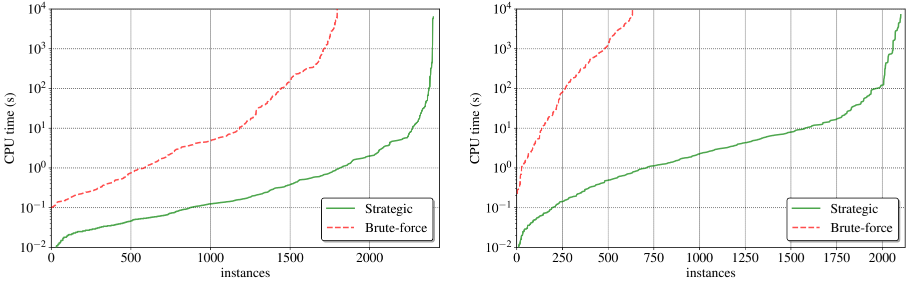
Experimential Evaluation

- Python implementation with reproducibility package.
- Experiments on a benchmark of real-world wide-area topologies (Topology Zoo) and three different datacenter network topologies (fat-tree, BCube and Xpander).
- Benchmarking: brute-force search vs. strategic search.
- Two hours timeout and 16 GB memory limit on AMD EPYC 7551 processors at 2.55 GHz.

Median Results

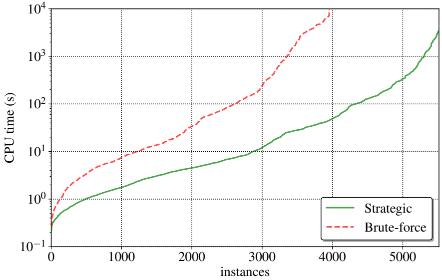
Topology	Problem	Brute-force (s)	Strategic (s)	Speedup
BCube	ON	79.5	1.7	47.1
BCube	OS	348.2	125.1	2.8
BCube	PS/PN	170.0	0.1	4684.0
Fat-tree	ON	59.4	1.2	47.6
Fat-tree	OS	2.0	0.2	8.5
Fat-tree	PS/PN	562.6	0.1	66976.3
Xpander	ON	407.3	3.0	137.7
Xpander	OS	124.1	1.6	78.0
Xpander	PS/PN	>7200.0	5.4	>1340.6
Topology Zoo	ON	59.6	4.6	12.9
Topology Zoo	OS	35.3	2.6	13.4
Topology Zoo	PS/PN	4.3	0.1	82.7

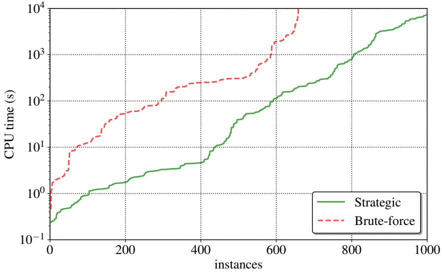
Pessimistic Scenario: ISP (left) Datacenter (right)



- Several orders of magnitude speed up.
- Datacenters have high path diversity and multiple shortest paths, and more opportunities for skipping of "uninteresting failures".

Optimistic Scenario: ISP (left) Datacenter (right)





- A few orders of magnitude speed up.
- Optimistic strategic search explores all the maximum failure scenarios, there are many of them in the highly connected datacenter topologies.

Conclusion

- Problem of finding feasible routings for multiple flows in network.
- Capacity-aware and accounting for multiple link failures.
- Introduced pessimistic and optimistic scenaria.
- Complexity overview for both splittlable and nonsplittable flows.
- Efficient strategic search algorithm and experiments.

Conclusion

- Problem of finding feasible routings for multiple flows in network.
- Capacity-aware and accounting for multiple link failures.
- Introduced pessimistic and optimistic scenaria.
- Complexity overview for both splittlable and nonsplittable flows.
- Efficient strategic search algorithm and experiments.

Future work

- Improve the performance of the optimistic strategic search.
- Study net reductions to reduce the number of minimum cuts.
- Parallel implementation.