Building Edge-Failure Resilient Networks

Chandra Chekuri

Bell Labs

Anupam Gupta

Bell Labs ! CMU

Amit Kumar

Cornell ! Bell Labs

Seffi Naor, Danny Raz

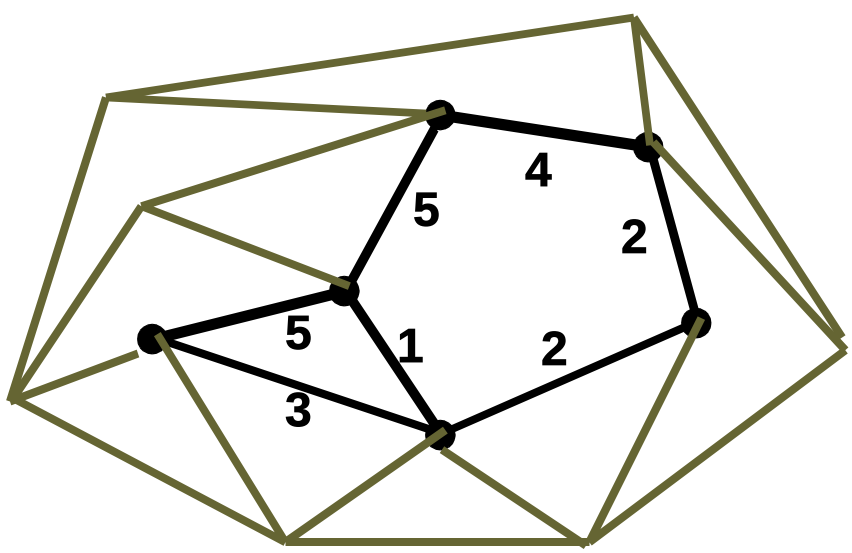
Technion

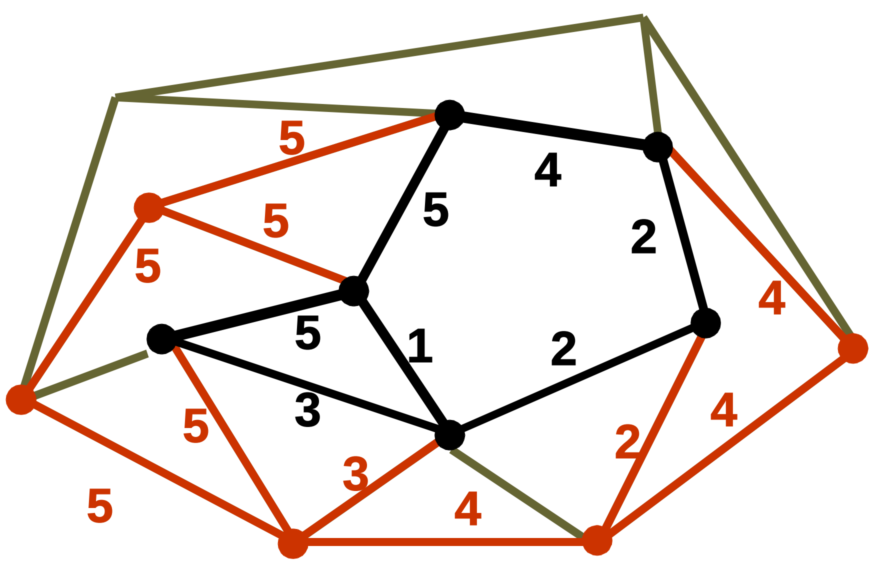
Paper in IPCO 2002

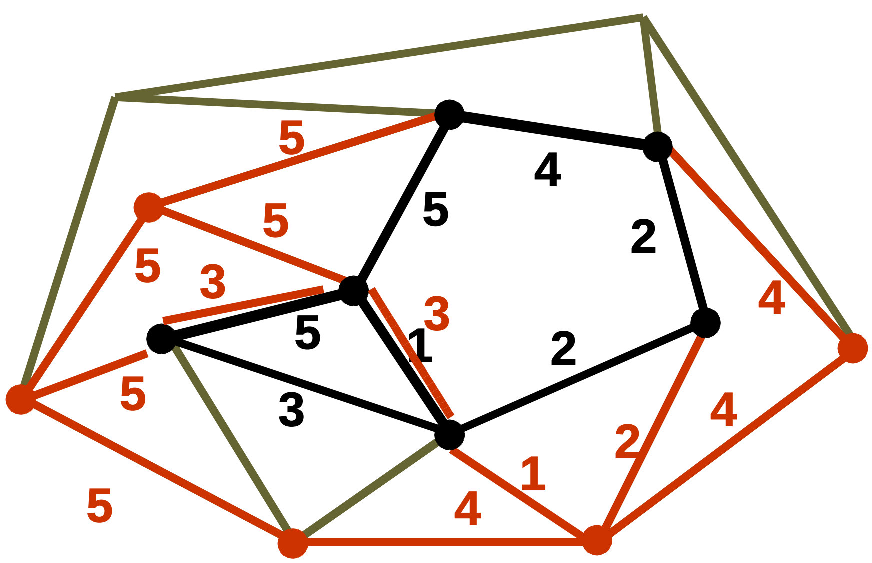
Problem

ï Given a “primary” network
Want to add a “backup network”

s.t. we can route even if an edge fails




5
5
5
5
4
2
5
1
2
4
4
5
3
2
3
5
4


5
4
5
5
2
5
3
3
4
5
1
2
5
3
4
5
2
1
4

Restoration Model

- Single/Multiple Edge failures
- Local restoration
- Cost linear/concave in capacity

Edge Failures

Assume:

At most *one* edge fails at any time

ï Commonly used assumption in practice.

ï Ideas/algorithms can be generalized to multiple edge failures

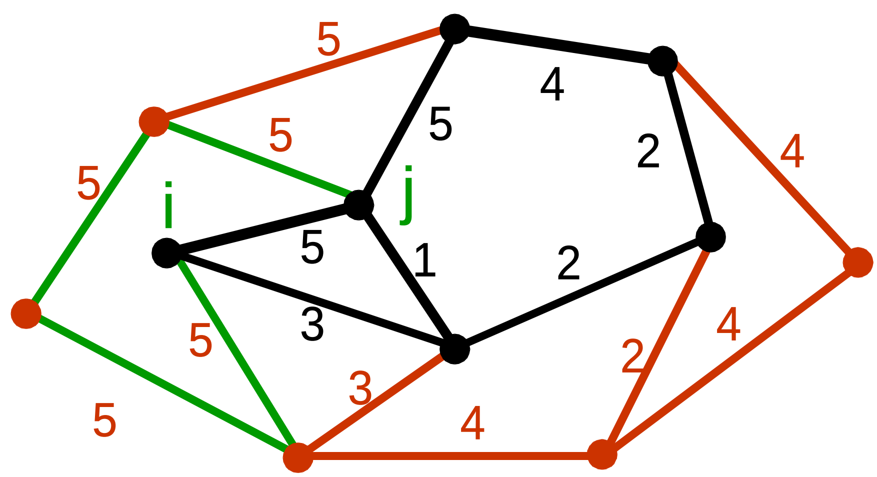
Local Restoration

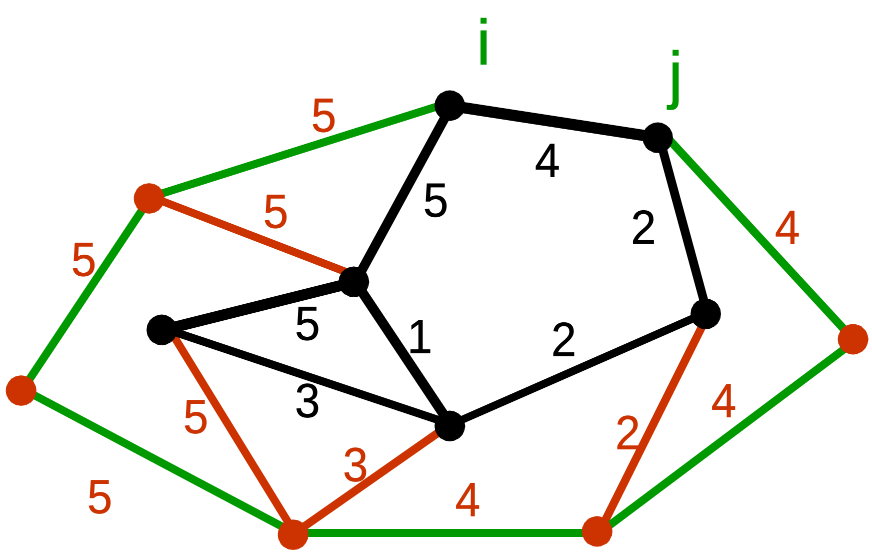
ï If $e = (i,j)$ with reservation u_e fails

Backup must be

ñ a single path $P(e)$ between i & j

ñ $P(e)$ has reservation at least u_e





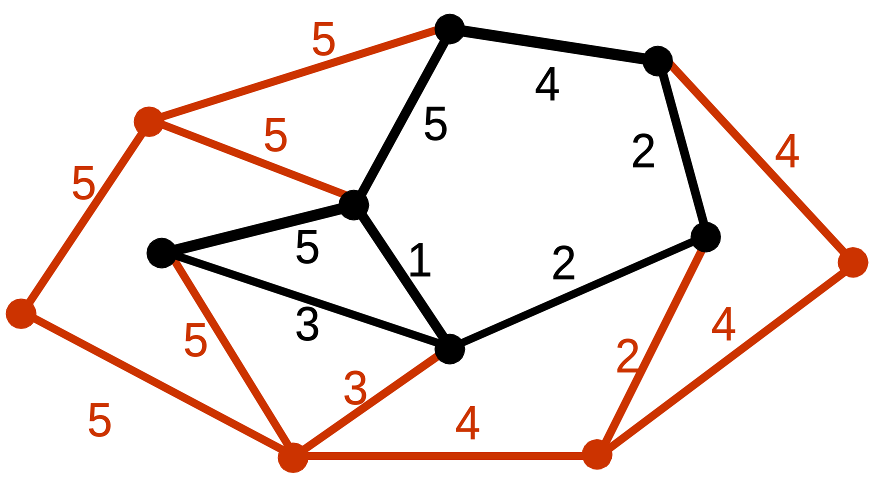
i
j
5
4
5
5
2
4
5
5
1
2
3
5
4
3
2
5
4

Multiplexing/Sharing

Since only one edge can fail

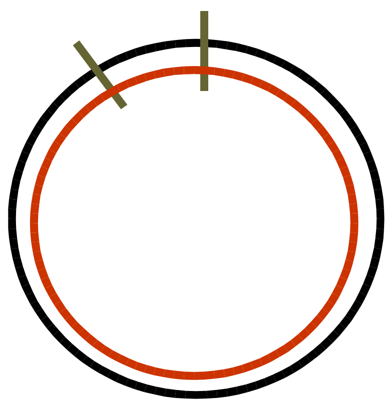
ñ Different backup paths can share same backup capacity reserved.

Cheaper than taking union of P(e)



Why Local Restoration?

- Quick and oblivious restoration (SONET Rings use local restoration)



- Single path for simplicity of routing (MPLS) and unsplittable demands.

Cost Model

Linear cost function: for edge e, cost c_e per unit bandwidth. Reserving capacity costs c_e w_e

ï Simplest case to understand.

ï In optical networks, capacity essentially unlimited, can buy additional capacity.

ï Algorithms work for concave costs.

ï Hard capacities result in difficult theoretical problems, might not be relevant in practice.

Backup Allocation

- Given:
  - Graph $G = (V,E)$
  - Edge costs c_e / unit bandwidth
  - Primary network A with reservation u_e
- Output:
  - Build backup network B s.t.

    for every edge e in A there is path with reservation u_e between endpoints of e in $B - \{e\}$
  - Cost = $\sum_e c_e w_e$

    where w_e is backup reservation

Provisioning & Backup

- Given:
  - Graph $G = (V,E)$
  - Edge costs c_e / unit bandwidth
  - Pair-wise demand matrix $D(i,j)$

- Output:
  - Primary network A to handle $D(i,j)$

    &

    Backup network B s.t.

    there is path of capacity u_e between endpoints of e in $B - \{e\}$

  - Cost = $\sum_e c_e (u_e + w_e)$

    where u_e is primary capacity

    w_e is backup capacity

Results

- Theorem #1

  A constant-factor approximation for the *Backup Allocation* problem for linear cost model and single edge failures.

- Theorem #2

  Given α approx for provisioning, an $O(\alpha \log n)$ approx for *P&B.*

  An $O(\log n)$ approximation algorithm for *Provisioning & Backup.*

Results (contd.)

- Theorem #3

  If primary network is tree T
  and want $T - \{e\} + P(e)$ also be tree

  - As hard as group Steiner problem on trees
  - If group Steiner tree on trees has an α approximation algorithm, we get an $O(\alpha)$ approximation.

Results (contd.)

Extensions

- For linear cost model $O(k)$ approximation for k edge failures.

- For concave costs $O(k \log u_{max}/u_{min})$ approximation. In terms of n, $O(k \log n)$ approximation.

Related Work

- Capacitated Survivable Network Design. Hard capacities, emphasis on inequalities to solve exactly. [Bienstock,Muratore], [Balakrishnan, Magnanti, Sokol, Wang]. Many others.

- Flow restoration instead of path restoration. [Brightwell,Oriolo,Shepherd], [Fleischer et al]

- Backup allocation for tree networks in VPN hose model [Italiano,Rastogi,Yener].

Our model slightly different from earlier ones.

Local restoration instead of end-to-end.

Goal – provably good approximation algorithms.

Backup Allocation

- Given:
  - Graph $G = (V,E)$
  - Primary network A with capacities u_e
  - Cost per unit bandwidth on e is c_e

- Output:
  - Backup network B such that

    For each edge $e = (i,j)\ 2\ A$

    there is path of capacity u_e between nodes i and j in $B - \{e\}$

- Objective: minimize cost of B

Suppose ...

- All capacities $u_e = 1$

  Want to build cheapest network B s.t.

  For each $(i,j) \in A$
  there is path between i and j in $B - \{e\}$

- **Steiner network problem:**

  Want to build cheapest network B s.t.

  For each $(i,j) \in A$

  there are r_{ij} edge-disjoint paths

  between i and j in B

- [Jain 98] 2-approximation algorithm for SN

Algorithm

- All capacities $u_e = 1$

- Algorithm:

(SN1)

$B_1 \leftarrow$ SN with $r_{ij} = 1$ for all $e = (i,j)$ 2 A.

For all $e = (i,j)$ 2 A

if e 2 B_1 then $r'_{ij} = 2$

else $r'_{ij} = 1$

(SN2)

Set cost of all edges in B_1 to 0

$B_2 \leftarrow$ SN with demands r'_{ij}

Output B_1 [B_2

Correctness

ï <u>Feasibility</u>

If $e \not\in B_1$ $\Rightarrow$ $B_1 - \{e\}$ has a path

If e 2 B_1 $\Rightarrow$ B_2 has two paths

$\Rightarrow$ $B_2 - \{e\}$ has a path

ï <u>Approximation Bound</u>

Opt(SN1) · OPT

$\Rightarrow$ cost(B_1) · 2 OPT

OPT [B_1 is feasible for SN2

(with cost OPT, due to zero costs)

$\Rightarrow$ cost($B_2 - B_1$) · 2 OPT

$\Rightarrow$ Approx bound of 4

An LP formulation

Min $\sum c_e \, w_e$

s.t.

w supports unit-flow between i & j in $E - \{e\}$

for all $e = (i,j)$ 2 A

Theorem:

The integrality gap of this LP is 4.

General u_e ?

ï Scaling + previous algorithm

ï Algorithm:

$A(k) = \{ e\ 2\ A \mid 2^k \cdot u_e < 2^{k+1} \}$

For all $e\ 2\ A(k)$, set $u_e = 2^{k+1}$

Independently for all k

Run previous algorithm on $A(k)$

Assign capacity 2^{k+1} on chosen edges $B(k)$

16 approximation solution. Can be improved to
$4e \approx 10.87$ approx.

Same algorithm works for concave costs,
approximation bound $O(\log u_{max}/u_{min})$

An LP formulation

Min $\sum c_e w_e$

s.t.

w supports u_e-flow between i & j in $E - \{e\}$

for all $e = (i,j)$ 2 A

<u>Theorem:</u>

The integrality gap of this LP is $\Theta(\log n)$.

Simultaneous Provisioning & Backup

Model?

ï How do we specify demands?

ï Two common models:

Point-to-point demand matrix D(u,v)
$\Rightarrow$ shortest-path routing optimal

VPN upper bounds
$\Rightarrow$ constant-factor approximation
[Gupta et al 01]

Possible other models...

Results

Theorem

Given α-approx algorithm for
provisioning in some model:

we get $O(\alpha \log n)$ approx algorithm for
backup & provisioning in that model

Hence:

$O(\log n)$-approx for Point-to-Point, VPN…

A two-step procedure

Algorithm:

Use provisioning algorithm to get A

(bandwidth allocation = u_e)

Use the previous backup algorithm acting on A to get backup network B

(bandwidth allocation = w_e)

Analysis

- Let u^* and w^* be an optimal solution OPT
- Claim:

  $u + u^* + w^*$ is a feasible solution for the LP
- Assuming this claim is true:

  $\text{cost}(A) \le \alpha\, \text{cost}(u^*) \le \alpha\, \text{OPT}$

  $\text{LP value} \le \text{cost}(A) + \text{cost}(u^* + w^*)$
  $\le (\alpha + 1)\, \text{OPT}$

  $\text{cost}(B) \le O(\log n)\, \text{LP} \le O(\alpha \log n)\, \text{OPT}$

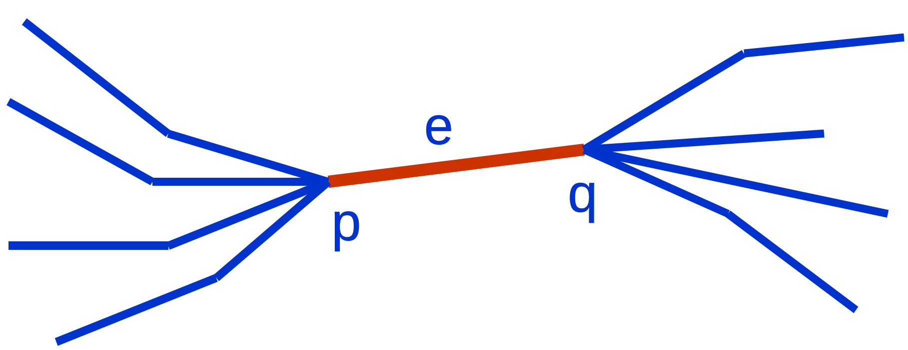
Analysis (contd.)

Proof Sketch (that $u + u^* + w^*$ is a valid LP sol'n):

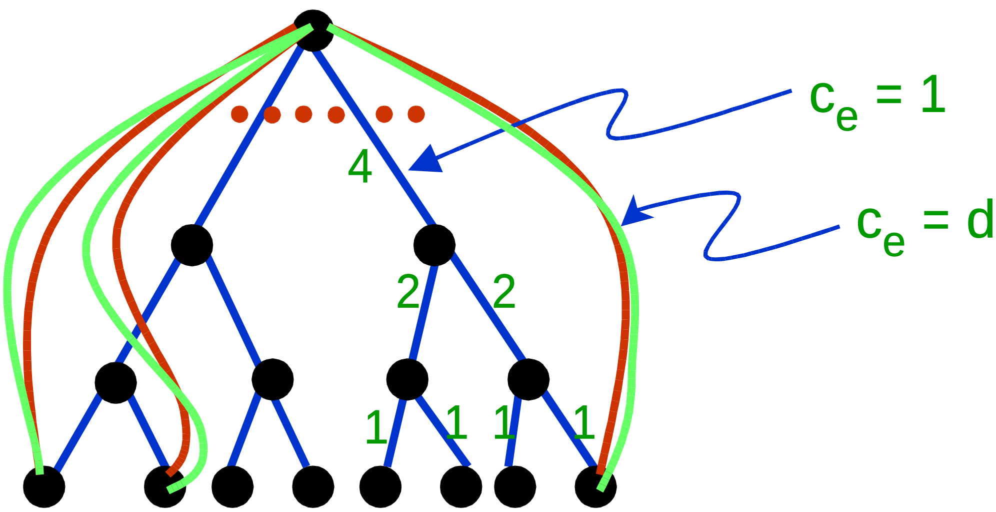
ï If $e = (p,q)$ goes down:

"Send back" u_e amount of flow using e back to the terminals using e

Now since $u^* + w^*$ forms a edge-failure resilient network: can use this to send "returned" flow in the desired fashion.



Tightness



ï Suppose $D(r, \text{leaf}) = 1$ for each leaf

Then step #1 can create the blue tree

From previous slide: OPT ¸ $\Omega(d^2\ 2^d)$

ï But if we chose green star instead

Red Star as backup costs $d2^d$

Future Work

- Improved approximations.

- Online models. Demand pairs arrive over time. Design primary and backup paths given existing primary and backup networks.

- Empirical evaluation.