Approximation Algorithms for Path Problems in Communication Networks

Thomas Erlebach (ETH Zürich)

❶ Maximum Edge-disjoint Paths Problem (MEDP)

❷ $O(\sqrt{m})$-approximation algorithm for MEDP

❸ $O(m^{0.5-\varepsilon})$ inapproximability of MEDP

❹ Unsplittable Flow Problem (UFP)

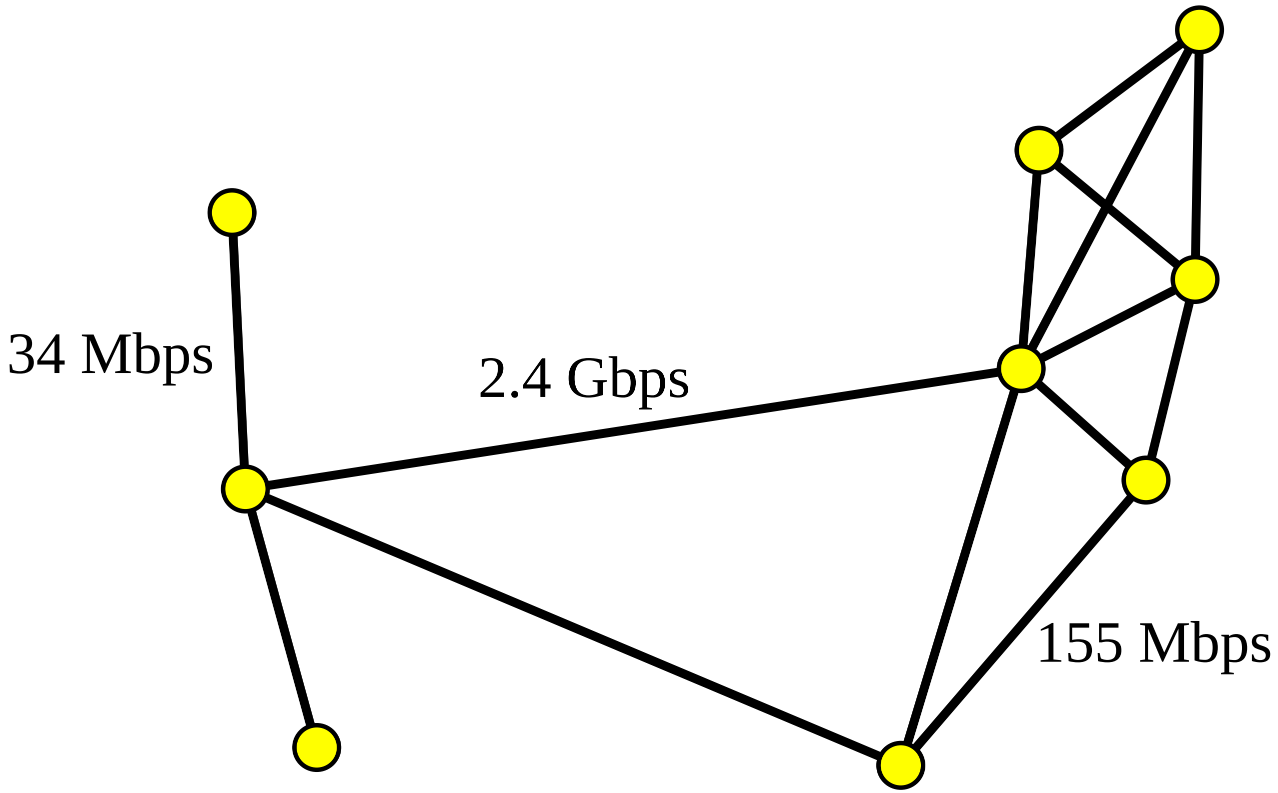
❺ $O(\sqrt{m})$-approximation algorithm for UFP

❻ $O(1)$-approximation for high-capacity UFP

❼ $O(1)$-approximation for MEDP in meshes

❽ Further known results and some open problems

Motivation: Bandwidth Reservation in Networks



The Maximum Edge-Disjoint Paths Problem (MEDP)

Instance:

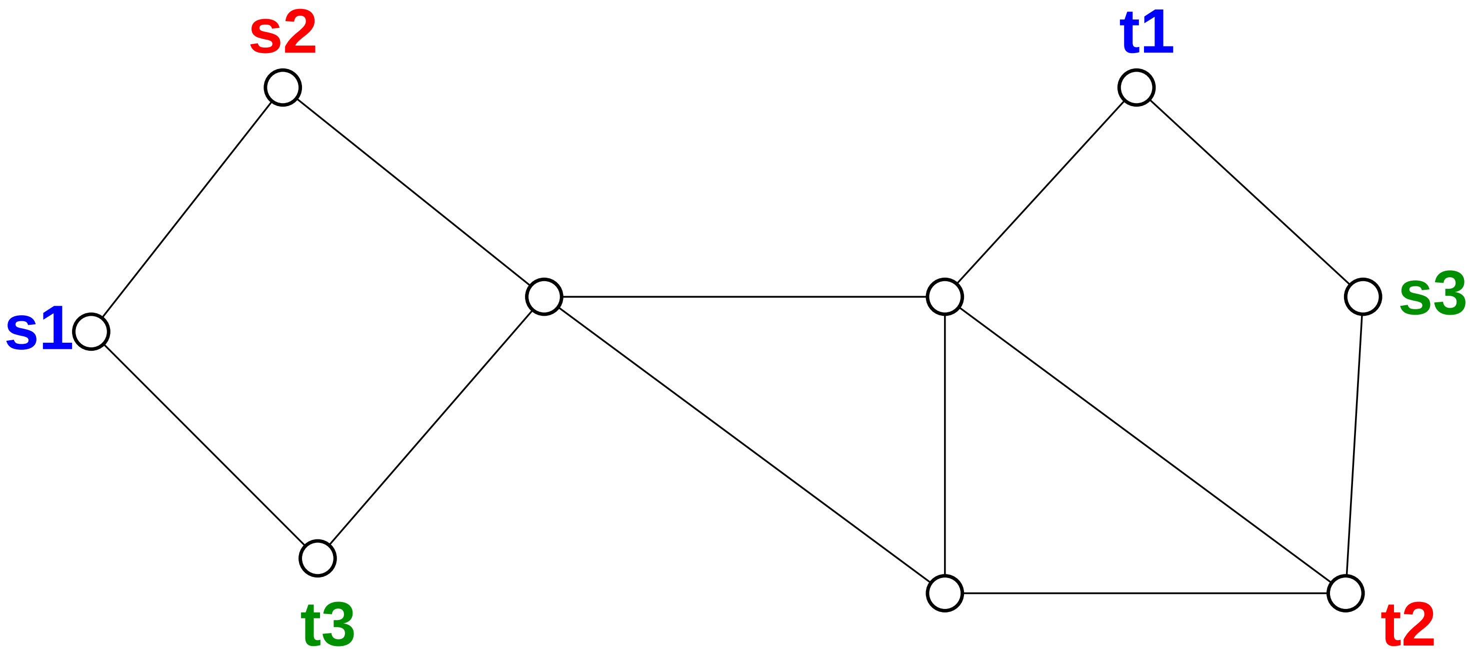
- graph $G = (V, E)$ with $|V| = n$ and $|E| = m$
- multi-set $\mathcal{T} = \{(s_i, t_i) \mid 1 \leq i \leq k\}$ of requests

Solution:

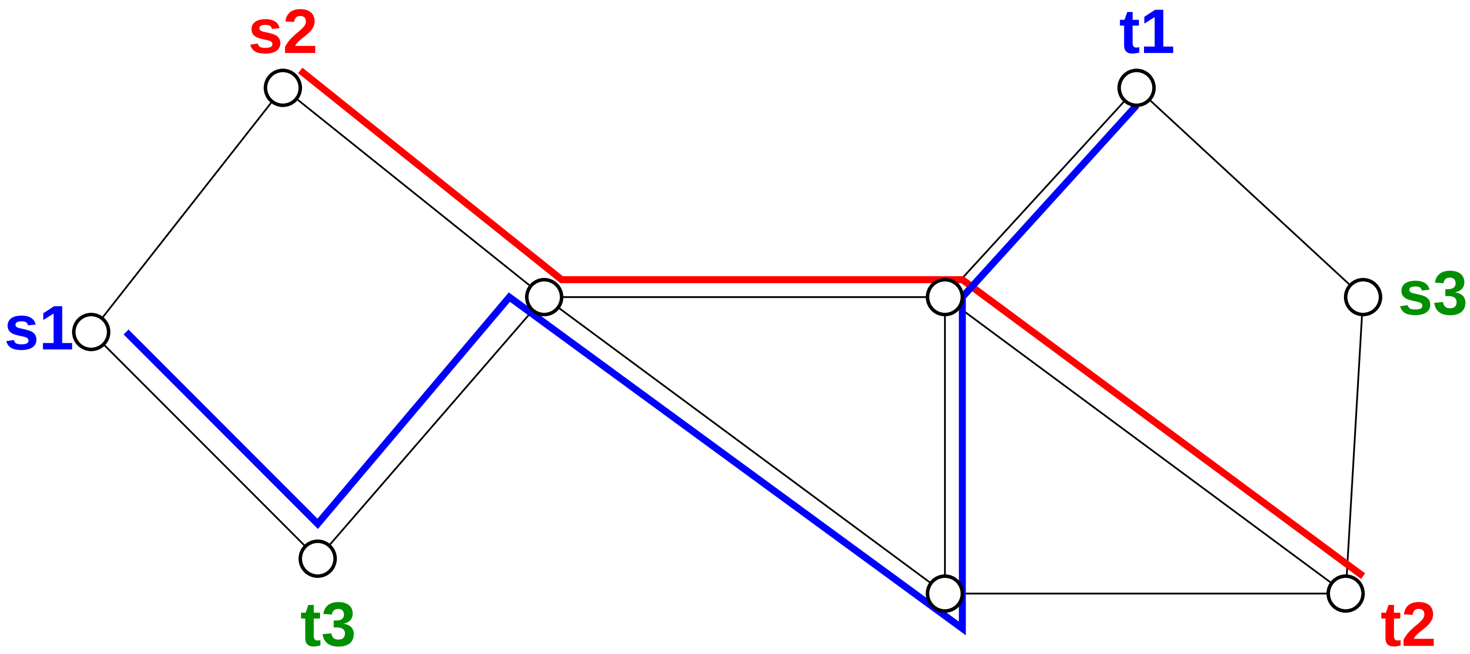
- subset $\mathcal{T}'$ of $\mathcal{T}$ and assignment of edge-disjoint paths to requests in $\mathcal{T}'$

Goal: maximize $|\mathcal{T}'|$

Example for MEDP

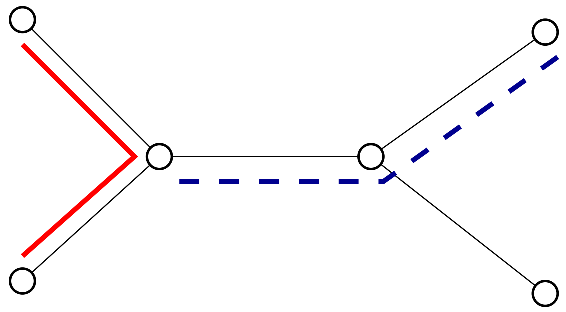


Solution to Example



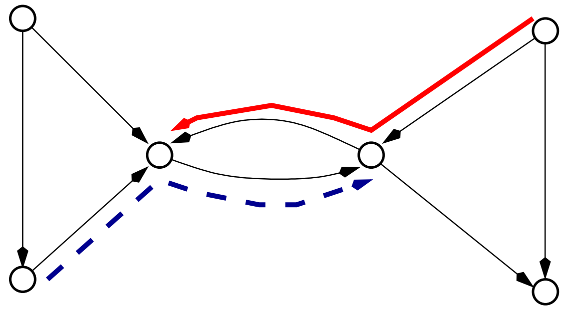
Variants of MEDP

★ undirected paths in undirected graphs

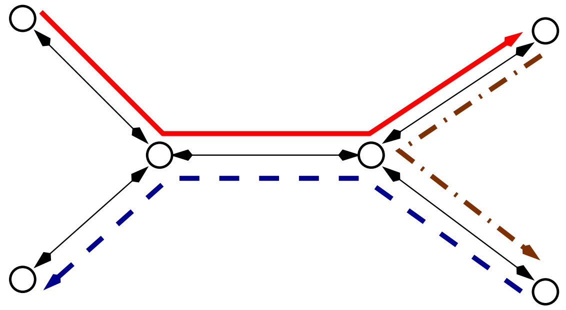


★ directed paths in directed graphs

(this is the hardest variant in general!)



★ directed paths in bidirected graphs



Definition: Approximation Algorithms for MEDP

OPT denotes the cardinality of an optimal solution.

An algorithm for MEDP is a ρ-approximation algorithm if it

- **runs in polynomial time**

  and

- **always outputs a solution $\mathcal{T}'$ with $|\mathcal{T}'| \geq \frac{OPT}{\rho}$.**

Complexity and Inapproximability of MEDP

- polynomial for chains, rings and stars
- polynomial for undirected trees, APX-hard for bidirected trees
- $\mathcal{NP}$-hard for meshes (Kramer and van Leeuwen, 1984)
- cannot be approximated within $O(m^{0.5-\varepsilon})$ for arbitrary directed graphs unless $P = \mathcal{NP}$ (Guruswami et al., 1999).
- polynomial for constant number of requests in undirected graphs (Robertson and Seymour), but $\mathcal{NP}$-hard even for only **two** requests in directed graphs (Fortune, Hopcroft, Wyllie, 1980)

The Shortest-Path-First Greedy Algorithm (SPFG)

$\mathcal{T}' \leftarrow \emptyset$;

while there exists a request in $\mathcal{T}$ that can still be routed **do**

$(s_i, t_i) =$ a request in $\mathcal{T}$ that can be routed using the fewest edges;

route (s_i, t_i) along a shortest path of available edges;

$\mathcal{T}' \leftarrow \mathcal{T}' \cup \{(s_i, t_i)\}$;

$\mathcal{T} \leftarrow \mathcal{T} \setminus \{(s_i, t_i)\}$;

od

Claim. SPFG is a $\sqrt{m}$-approximation algorithm.

Analysis of SPFG (Kolliopoulos and Stein, 1998)

- Compare solution of SPFG to some optimal solution S^*, $|S^*| = OPT$.
- When SPFG accepts a request along a path p, remove all paths intersecting p from S^*.

Let $m_o \leq m$ be the number of edges used by paths in S^*.

- While SPFG accepts paths that are shorter than $\sqrt{m_o}$, each accepted path intersects at most $\sqrt{m_o}$ paths from S^*.
- When SPFG starts to consider paths of length at least $\sqrt{m_o}$, all remaining paths in S^* have length at least $\sqrt{m_o}$ and there can be at most $m_o/\sqrt{m_o} = \sqrt{m_o}$ of them.
- Solution of SPFG contains at least $OPT/\sqrt{m_o}$ paths.

Analysis of SPFG (Version 2)

Claim. SPFG outputs a solution of size $\Omega\left(\frac{OPT^2}{m_o}\right) = \Omega\left(\frac{OPT}{\frac{m_o}{OPT}}\right)$.

Proof. Assume SPFG accepts t paths $p_1, p_2, \ldots, p_t$.

$k_i :=$ number of paths removed from S^* because of p_i (except p_i)

➠ p_i has length at least k_i.

➠ The k_i paths removed from S^* because of p_i have length at least k_i and use at least k_i^2 edges in total.

$$\Rightarrow \quad m_o \geq \sum_{i=1}^{t} k_i^2 \geq \frac{\left(\sum_{i=1}^{t} k_i\right)^2}{t} \approx \frac{OPT^2}{t}$$

↑ (Cauchy-Schwarz)

□

Inapproximability of MEDP

Theorem. MEDP in directed graphs is $\mathcal{NP}$-hard to approximate within $O(m^{0.5-\varepsilon})$. (Guruswami, Khanna, Rajaraman, Shepherd, Yannakakis, 1999)

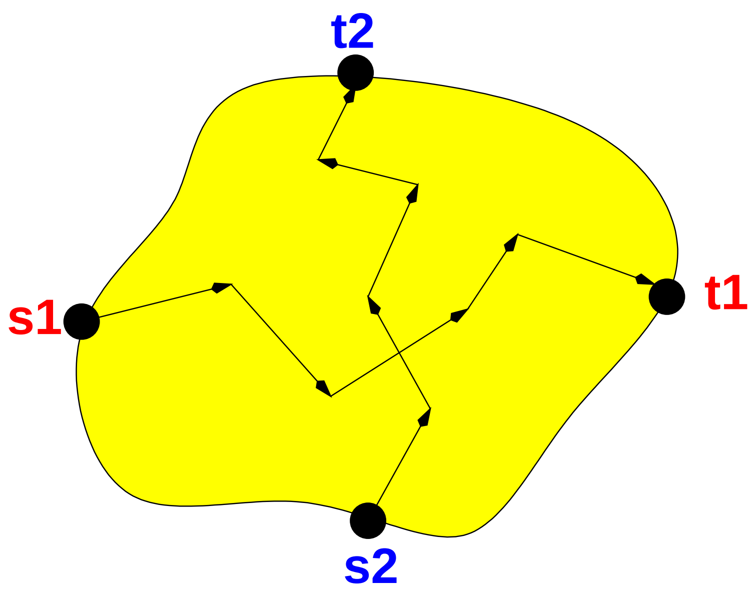
Proof. By reduction from 2DIRPATH.

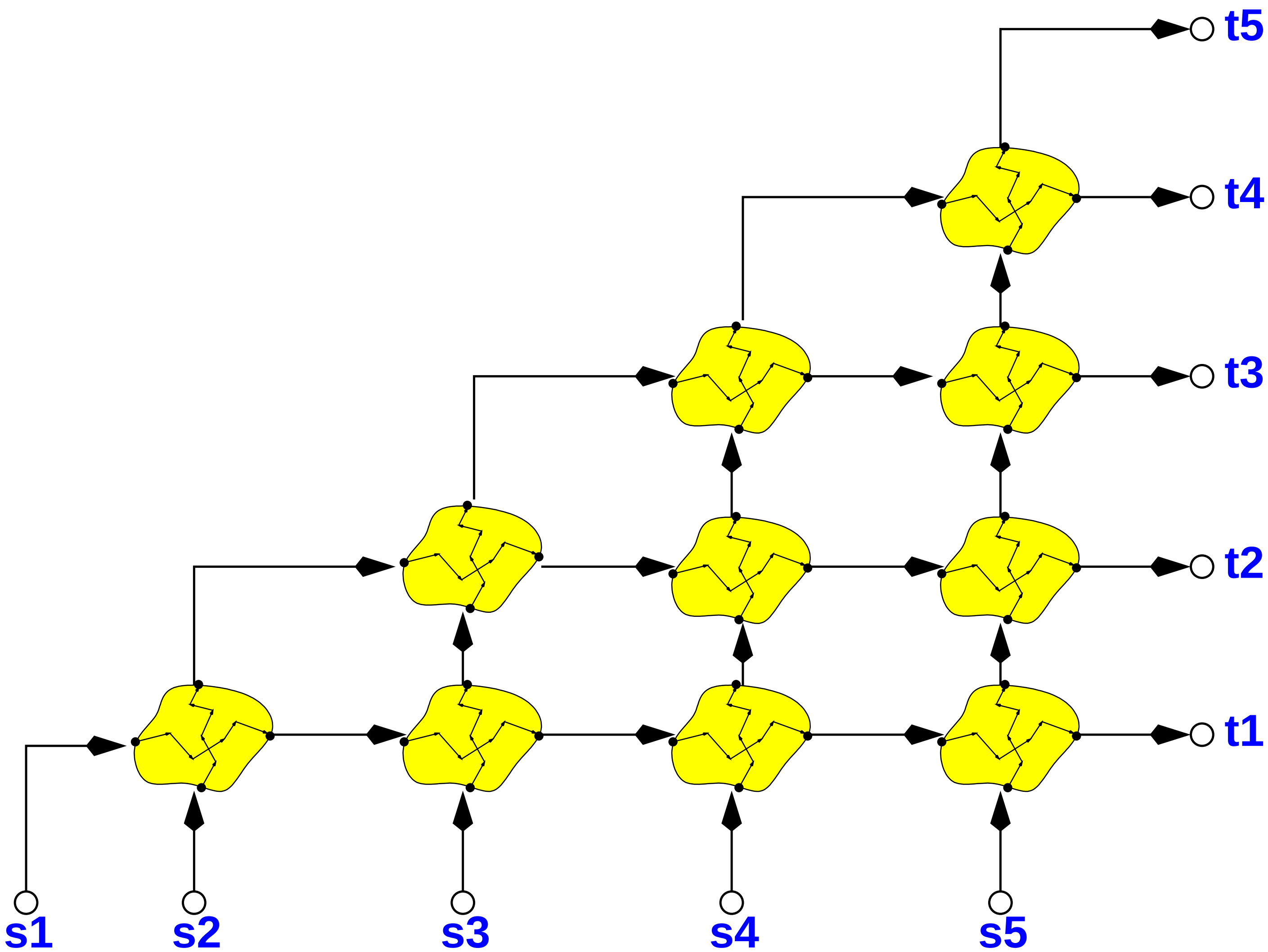
2DIRPATH:

Given: directed graph $H = (V, A)$

Question: are there 2 edge-disjoint paths from s_1 to t_1 and s_2 to t_2?

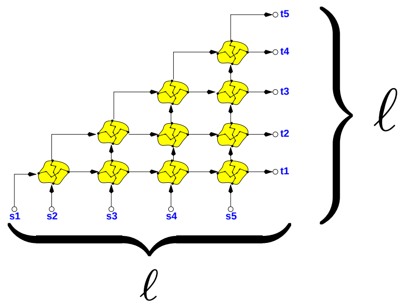
2DIRPATH is $\mathcal{NP}$-complete




t5
t4
t3
t2
t1
s1
s2
s3
s4
s5

❶ Choose $\ell = |A|^{1/\varepsilon}$ for some constant $\varepsilon > 0$.

❷ Apply construction for ℓ requests:



> If H is a YES-instance, $OPT = \ell$.
>
> If H is a NO-instance, $OPT = 1$.

➥ Resulting graph has $m = \Theta(\ell^2 |A|) = \Theta(\ell^{2+\varepsilon})$ edges.

➥ approximating MEDP with ratio $\ell = m^{\frac{1}{2+\varepsilon}} = m^{0.5-\varepsilon'}$ is $\mathcal{NP}$-hard. $\square$

The Unsplittable Flow Problem (UFP)

Instance:

- graph $G = (V, E)$ with edge capacities $u(e) \in \mathbb{R}$
- multi-set $\mathcal{T} = \{(s_i, t_i, d_i, r_i) \mid 1 \leq i \leq k\}$ of requests

$d_i =$ **demand** of request i

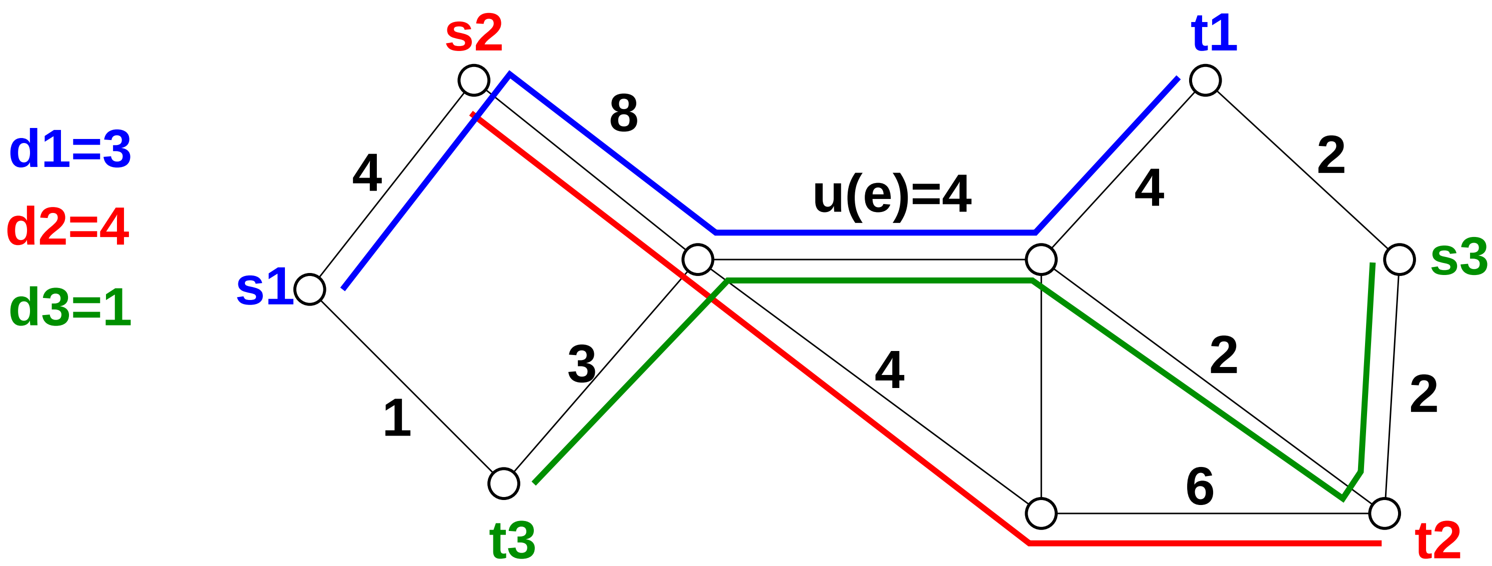
$r_i =$ **profit** of request i

Solution:

- subset $\mathcal{T}'$ of $\mathcal{T}$ and assignment of paths to requests in $\mathcal{T}'$ such that no edge capacity is exceeded

Goal: maximize the total profit $\sum_{i \in \mathcal{T}'} r_i$

Example of unsplittable flow



Variants of UFP

$d_{\max}$ = largest demand

$u_{\min}$ = minimum edge capcity

- **Classical UFP:** $d_{\max} \leq u_{\min}$
  - any request can be routed through any edge
- **Extended UFP:** $d_{\max}$ can be arbitrary
  - it may be impossible to route some requests through certain edges
- **Bounded UFP:** $d_{\max} \leq \frac{1}{K} u_{\min}$
  - at least K requests can be routed through any edge

An Approximation Algorithm for Classical UFP

(Azar and Regev, 2001)

❶ **Separate the big requests and the small requests.**

Partition $\mathcal{T}$ into $\mathcal{T}_1$ and $\mathcal{T}_2$:

- → $\mathcal{T}_1$ consists of requests with $d_i \leq \frac{1}{2}u_{\min}$
- → $\mathcal{T}_2$ consists of requests with $d_i > \frac{1}{2}u_{\min}$

➪ Compute solutions for $\mathcal{T}_1$ and $\mathcal{T}_2$ separately.

➪ Output the better of the two solutions.

➥ This loses at most a factor of 2 in the approximation ratio.

❷ Consider the gained profit relative to the added load.

For request j and a path P from s_j to t_j define:

$$F(j,P) = \frac{r_j}{\sum_{e \in P} \frac{d_j}{u(e)}}$$

Idea: Accept request j if $F(j,P)$ is above some threshold α.

We have:

$$\alpha_{\min} := \frac{r_{\min}}{n} \leq F(j,P) \leq \frac{r_{\max} u_{\max}}{d_{\min}} =: \alpha_{\max}$$

➥ Try all powers of 2 between $2^{\lfloor \log \alpha_{\min} \rfloor}$ and $2^{\lceil \log \alpha_{\max} \rceil}$ as possible values for the threshold α, and take the best solution.

❸ **Algorithm for set S (either $S = \mathcal{T}_1$ or $S = \mathcal{T}_2$) and threshold α.**

for $j \in S$ in order of non-increasing $\frac{r_j}{d_j}$ **do**

  if $\exists$ valid path P from s_j to t_j with $F(j, P) > \alpha$ **then**

    accept request j and route it on path P;

  else

    reject request j;

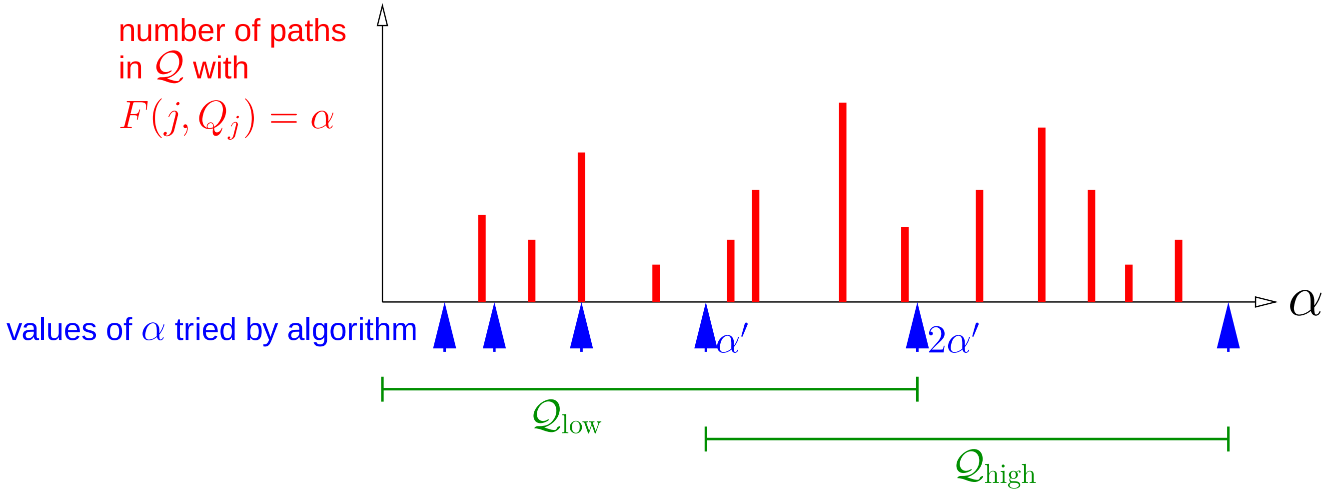
  fi

od

Path P is **valid** for request j if it can be routed along P without violating any edge capacity.

Analysis of the algorithm

- Consider optimal solution $\mathcal{Q}$ for $\mathcal{T}_1$ (or for $\mathcal{T}_2$)
- $Q_j :=$ path assigned to request $j \in \mathcal{Q}$
- Consider distribution of $F(j, Q_j)$ for $j \in \mathcal{Q}$:



Consider α' with $r(\mathcal{Q}_{\text{low}}) \geq \frac{1}{2} r(\mathcal{Q})$ and $r(\mathcal{Q}_{\text{high}}) \geq \frac{1}{2} r(\mathcal{Q})$.

Claim. For $\alpha = \alpha'$ the algorithm yields an $O(\sqrt{m})$-approximation.

$\mathcal{P}$:= set of requests routed by the algorithm (when called with $\mathcal{T}_i$ and α')

E_{heavy} := edges with load $\geq \frac{1}{4}$ at the end of the algorithm

Case 1: $|E_{\text{heavy}}| \geq \sqrt{m}$.

Can show:

$$\begin{aligned} r(\mathcal{Q}_{\text{low}}) &\leq 2m\alpha' \\ r(\mathcal{P}) &\geq \frac{1}{4}\sqrt{m}\alpha' \end{aligned}$$

Case 2: $|E_{\text{heavy}}| < \sqrt{m}$.

Can show: $$r(\mathcal{Q}_{\text{high}} \setminus \mathcal{P}) \leq 4\sqrt{m} \cdot r(\mathcal{P})$$

$\square$

Making the algorithm strongly polynomial

The running-time of the algorithm is polynomial, but depends on the logarithm of numbers in the input: $\log \frac{n \cdot r_{\max} \cdot u_{\max}}{r_{\min} \cdot d_{\min}}$ values of α are tested.

Recall that $k :=$ number of requests.

- if $u(e) > k \cdot d_{\max}$, set $u(e) = k \cdot d_{\max}$
- throw away requests with $r_j < \frac{1}{k} r_{\max}$ ➠ we get $\frac{r_{\max}}{r_{\min}} \leq k$
- treat "tiny" requests (with $d_j \leq \frac{1}{k} u_{\min}$) separately

➥ Resulting algorithm has ratio $O(\sqrt{m})$ and is strongly polynomial.

Further Results for Unsplittable Flow

(Azar and Regev, 2001)

- **Extended UFP**:
  - approximation ratio $O\left(\sqrt{m} \cdot \log\left(2 + \frac{d_{\max}}{u_{\min}}\right)\right)$
  - $m^{1-\varepsilon}$-inapproximability for directed graphs
  - $m^{0.5-\varepsilon}\sqrt{\lfloor \log \frac{d_{\max}}{u_{\min}} \rfloor}$-inapproximability for directed graphs
- **Bounded UFP** ($d_{\max} \leq \frac{1}{K} u_{\min}$):
  - approximation ratio $O(K \cdot n^{1/K})$ for $K \geq 2$ (works also on-line!)

The High-Capacity Case of Unsplittable Flow

(Guruswami et al., 1999)

☆ Formulate UFP as an Integer Linear Program (ILP).

☆ Solve LP relaxation optimally.

☆ Use **randomized rounding** (Raghavan and Thompson, 1987) to get an integer solution.

If $d_{\max} \leq \frac{u_{\min}}{c \log m}$ for some sufficiently large constant c, then there is an $O(1)$-approximation for UFP.

An $O(1)$-Approximation Algorithm for Meshes

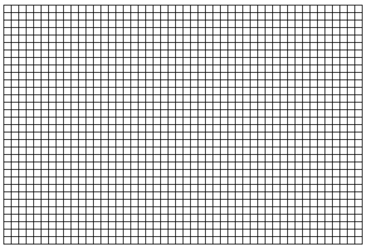
(Kleinberg and Tardos, 1995)

❶ Partition the mesh into submeshes of size $\gamma \log n \times \gamma \log n$.

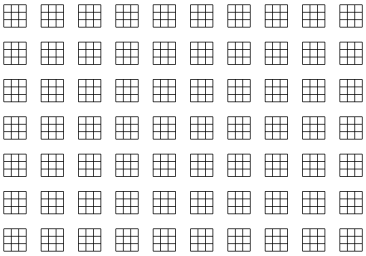
❷ Choose random subset of submeshes with mutual distance $\geq 2\gamma \log n$.

❸ Consider short requests and long requests separately and take the better of the two solutions.

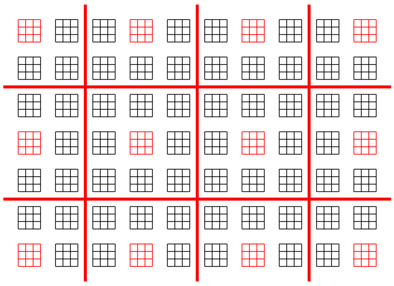
The mesh:



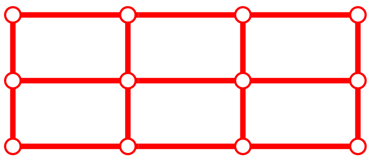
Partitioning into submeshes:



Randomly selected submeshes:



Simulated network with edge capacities $\Omega(\log n)$:



Handling of long requests (distance $> 16\gamma \log n$):

- Use randomized rounding in simulated network.
- Translate accepted paths back into the mesh.

Handling of short requests (distance $\leq 16\gamma \log n$):

- Apply algorithm recursively within selected submeshes.
- Long requests of recursive call are handled as above.
- Short requests of recursive call: brute-force.

➥ **approximation ratio $O(1)$ for meshes**

Further Known Results (1)

- MEDP in random graph $G_{n,p}$ with average degree $d \geq \ln n$:
  w.h.p., can route **all** requests in any request set of cardinality $O(\frac{m}{\log_d n})$
  (Broder, Frieze, Suen and Upfal, 1994)

- MEDP in random r-regular graph (r sufficiently large constant):
  w.h.p., can route **all** requests in any request set of cardinality $O(\frac{rn}{\log_r n})$
  (Frieze and Zhao, 1999)

- Edge-expansion $\beta(G) = \min_{S \subseteq V : |S| \leq n/2} \frac{|\delta(S)|}{|S|}$ and max. degree Δ
  ➥ approximation ratio $O(\Delta^2 \beta^{-2} \log^3 n)$ for UFP with uniform capacities (Srinivasan, 1997; Kleinberg and Rubinfeld, 1996)
  - ratio $O(\text{polylog } n)$ for butterfly and related networks

Further Known Results (2)

- ratio $(\frac{5}{3} + \varepsilon)$ for MEDP in bidirected trees (E. and Jansen, 1998)
- ratio $O(1)$ for MEDP in complete graphs (E. and Vukadinović, 2001)
- ratio $O(1)$ for MEDP in trees of rings (E., 2001)
- **Maximum path coloring**:

  given W colors, can accept W sets of edge-disjoint paths.

  Reduction: ratio ρ for MEDP ➨ ratio $\frac{1}{1-e^{-1/\rho}} < \rho + 1$ for MaxPC

  (Awerbuch et al., 1996)
- **Online algorithms** (preemptive/non-preemptive, deterministic/randomized)

Problem Variants and Related Problems

- Single-source unsplittable flow (Kolliopoulos & Stein, 1997; Dinitz, Garg & Goemans, 1999; Skutella, 2000)
- Integral splittable flow (Guruswami et al., 1999)
- Bounded-length edge-disjoint paths (Guruswami et al., 1999)
- Routing in rounds, path coloring, call scheduling, congestion minimization

Some Open Problems

★ (In-)approximability of MEDP in undirected graphs.
(Known: APX-hard, $O(\sqrt{m})$-approximation)

★ (In-)approximability of half-disjoint paths problem or UFP with $d_{\max} \leq \frac{u_{\min}}{2}$.
(Known: $\mathcal{NP}$-hard, $O(\sqrt{n})$-approximation)

★ Find better algorithms for MEDP and UFP in restricted classes of graphs that include realistic topologies.
(For example: partial k-trees)

References

- Jon Kleinberg. **Approximation Algorithms for Disjoint Paths Problems.** PhD Thesis, MIT, 1996.
- Stavros Kolliopoulos and Clifford Stein. **Approximating Disjoint-Path Problems Using Greedy Algorithms and Packing Integer Programs.** IPCO VI, 1998.
- Venkatesan Guruswami, Sanjeev Khanna, Rajmohan Rajaraman, Bruce Shepherd, and Mihalis Yannakakis. **Near-Optimal Hardness Results and Approximation Algorithms for Edge-Disjoint Paths and Related Problems.** STOC, 1999.
- Yossi Azar and Oded Regev. **Strongly Polynomial Algorithms for the Unsplittable Flow Problem.** IPCO VIII, 2001.
- Jon Kleinberg and Eva Tardos. **Disjoint paths in densely embedded graphs.** Proc. 36th FOCS, 1995.