Algorithmic Challenges in Optical Network Design

Chandra Chekuri
Univ. of Illinois (UIUC)

Lisa Zhang
Bell Labs

Modern Optical Networks

Signals/data transmitted as light on optical fiber

- Very high capacity
- Based on DWDM technology
- Ultra long haul
- Mesh based (as opposed to older ring based networks such as SONET)

Pros: capacity and speed required for modern networks

Challenges: recent and sophisticated technology (brittle), high cost, *optimization/verification*

Three Key Optical Technologies

1. Wavelength Division Multiplexing

Dense Wavelength Division Multiplexing (DWDM)

100+ wavelengths per fiber; 10Gbps/λ; 1 Tbps per fiber

What is a terabit?

60,000,000 text page; 200,000 photographs, 40,000 music files; 25 movie videos

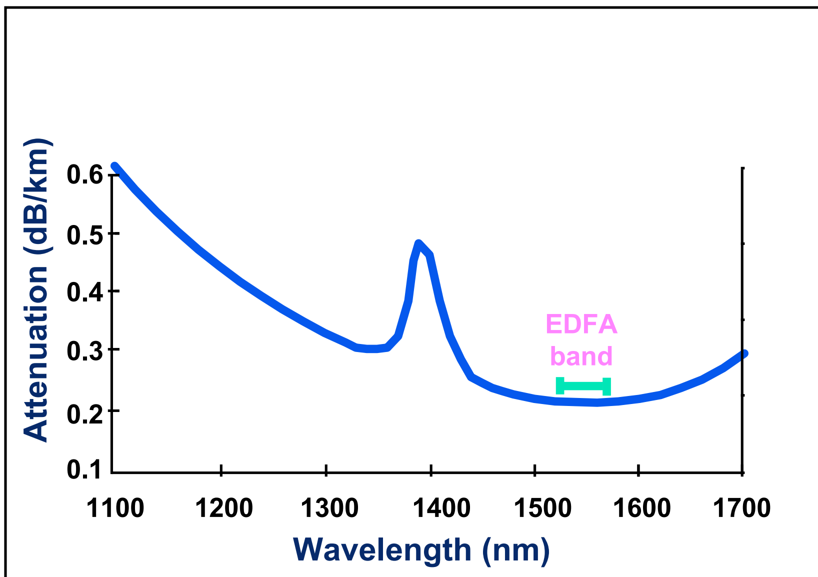
4960 hours at 56 kilobits/second (telephone modem)
278 hours at 1 megabit/second (cable modem)
17 minutes at 1 gigabit/second (gigabit ethernet)

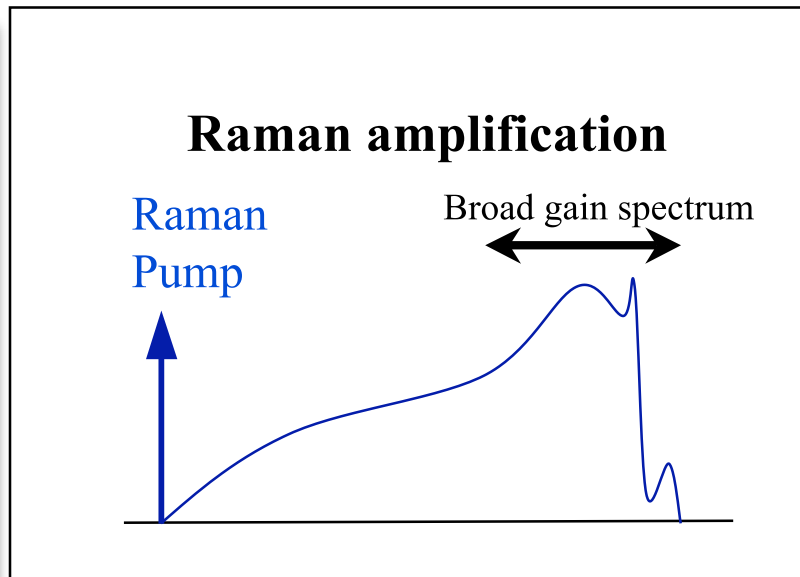
Three Key Optical Technologies

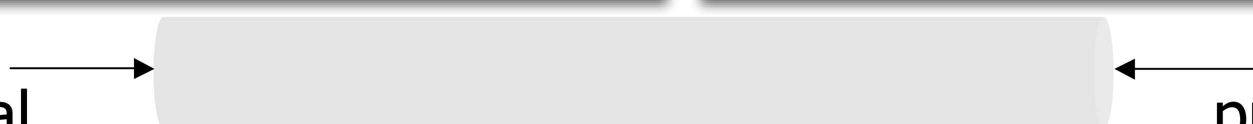
2. Optical (Raman) Amplification

Signals travel long distance (>1000 km) within optical domain

Wavelengths simultaneously amplified (non-linear problem)







data signal

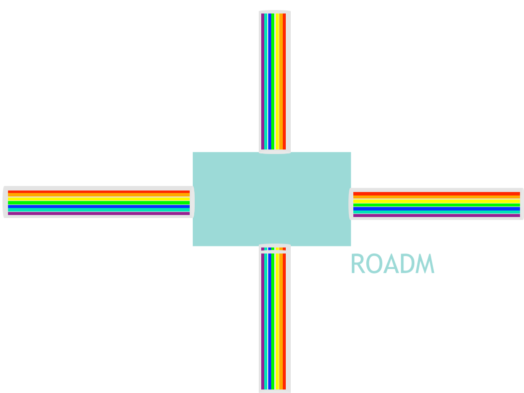
pump signal

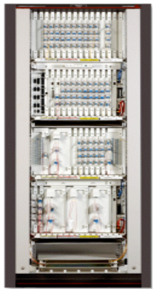
Three Key Optical Technologies

3. Wavelength granular optical switching

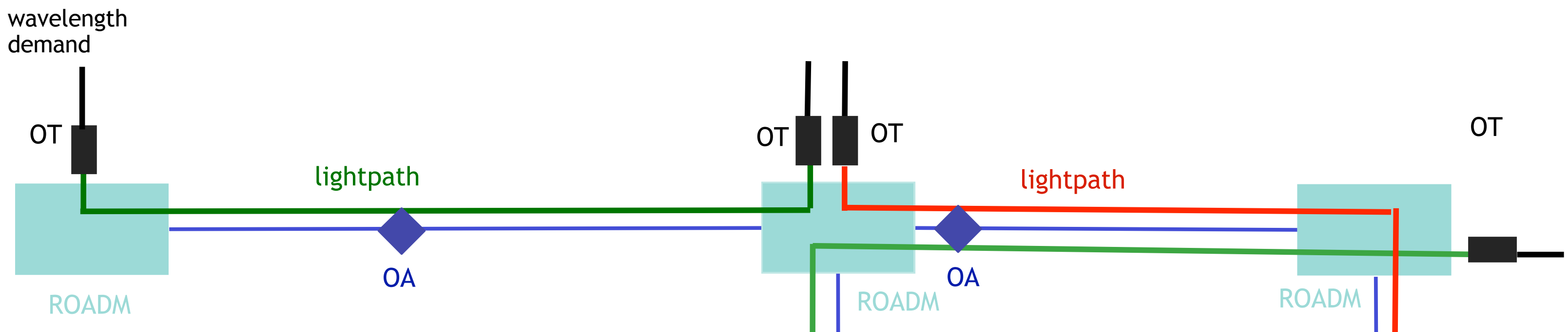
Allows single-wavelength light path travel in network without O-E-O at any intermediate network element.

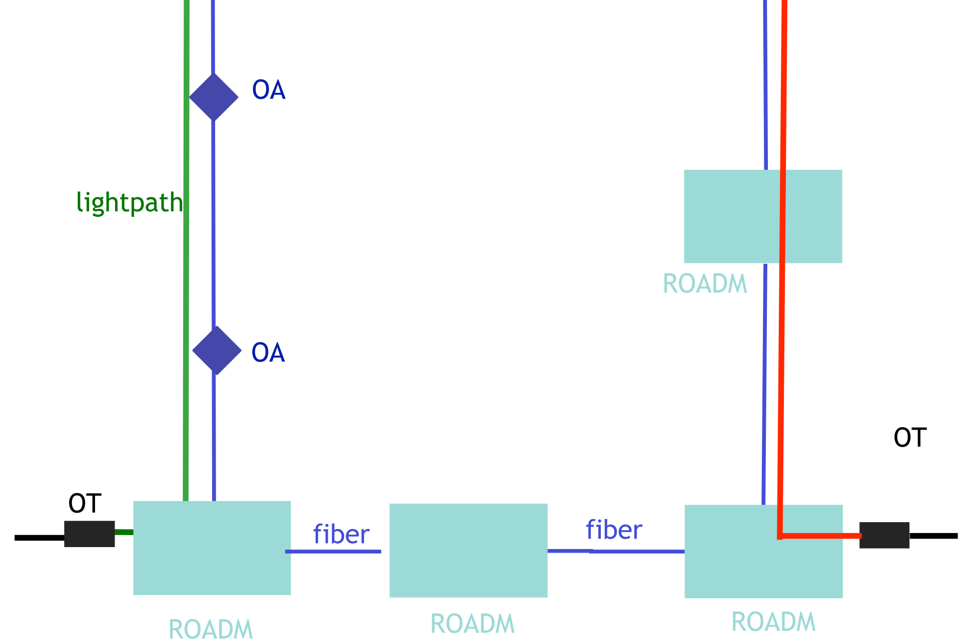
Accomplished by Reconfigurable Optical Add/drop multiplexer (ROADM)





Optical Components





ROADM: Routes each wavelength individually

OT: optical transponder. Performs O-E, E-O, translates to particular wavelength

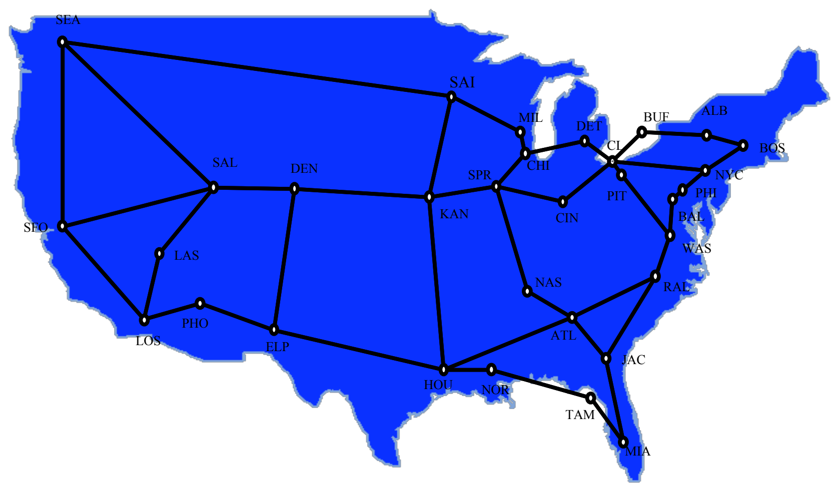
OA: optical amplification.

Strengthens OSNR, offsets dispersion, etc.

Costs:

ROADM : OT: OA ~ 10: 1: 5

Design Problem

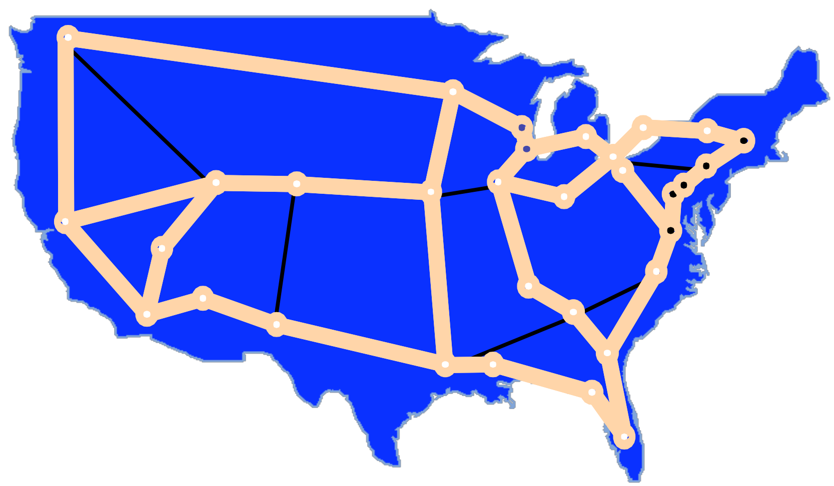


Goal: build an optical backbone network

Traffic: estimates of demands between major metros

Dark fiber: network where fiber is in the ground

Design Problem



Goal: install equipment on network (light up some fibers in dark network) to satisfy (route) traffic

Objectives: minimize cost, maximize fault tolerance and expandability, and ...

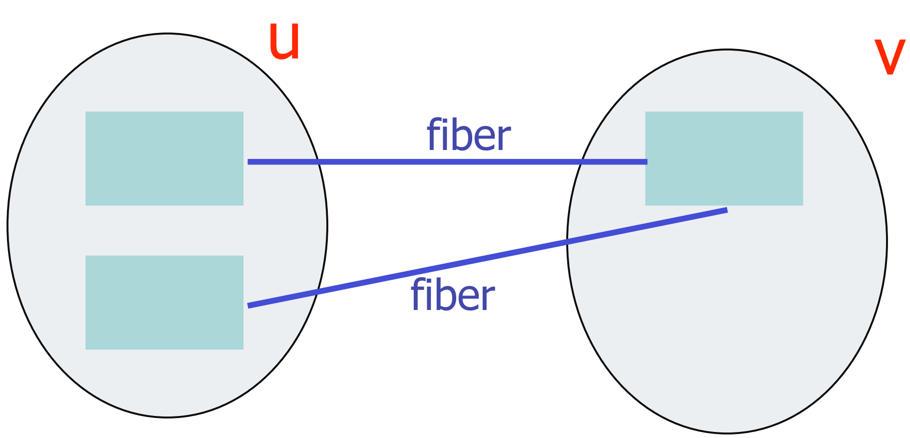
Input in more detail

- Dark fiber network: graph $G=(V,E)$
- Traffic: granularity of a single wavelength
  - source-destination pairs: $s_1t_1, s_2t_2, \ldots, s_ht_h$
  - for each pair s_it_i a *protection requirement* (more later)

- Equipment information
  - ROADM types, OT types, ...
  - Constraints on equipment (usually messy)
- Cost for various equipment:
  - ROADM, OT, OA, fiber, circuit packs, ...
- Reach and regeneration constraints (physics)
  - upper bound on distance before need for OA
  - number of optical devices before regeneration (OT)

What is a feasible solution?

- For each edge e, k_e the number of fibers on e
- For each node v, k_v the number of ROADMS at v
- If $e = (u, v)$ which fiber on e is connected to which ROADM at u and which ROADM at v



What is a feasible solution?

For each demand $s_i t_i$

- a sequence of ROADMs (at nodes) and fibers (on edges)
- on each fiber the wavelength
- OT locations for wavelength conversion and regeneration

Very complicated and difficult to optimize

Break Problem into Tractable Pieces

- Buy-at-Bulk Network Design
  - Choose # of fibers per edge and *routing* for each demand
- Assign fibers and wavelengths to each demand (note that route is already fixed)

Alternative: combine above two steps into one step

- ROADM assignment at nodes and connection of fibers
- OT assignment for reach and wavelength conversion
- Check physical level constraints and iterate

Protection Constraints

Fault-tolerance very important in high-capacity networks

Potential failures:

- fiber cut
- equipment failure (OA, OT, ROADM)
- power failure at a node location etc

Remedy: 1+1 protection

For each demand $s_i t_i$ choose two paths P_i (primary) and Q_i (backup)

- P and Q are internally node/link/fiber disjoint
- route data along both paths *simultaneously*

Ring networks (SONET) provided protection implicitly/automatically.

For new mesh networks, part of optimization

Outline for rest of the talk

- Approximation algorithms for buy-at-bulk network design - a survey [Chandra]
- Experience with some heuristics on buy-at-bulk for optical network design [Lisa]
- Wavelength assignment problems/issues [Lisa]

Buy-at-Bulk Network Design

Undirected graph $G=(V,E)$

for each E, edge cost function $f_e: \mathcal{R}^+ \rightarrow \mathcal{R}^+$

Demand pairs: $s_1t_1, s_2t_2, \ldots, s_kt_k$

Demands: s_it_i has a positive demand d_i

Feasible solution: for each pair s_it_i, a path P_i connecting s_i and t_i along which d_i flow is routed

Cost of flow: $\sum_e f_e(x_e)$ where x_e is the cumulative flow on e

Goal: minimize cost of flow

Special case: Single-source BatB

source s, terminals t_1, t_2, ..., t_k
demand d_i from s to t_i

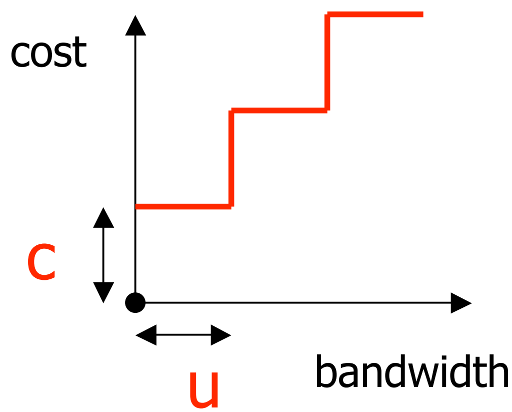
general case: multi-commodity

What is the cost function?

Optical networks:

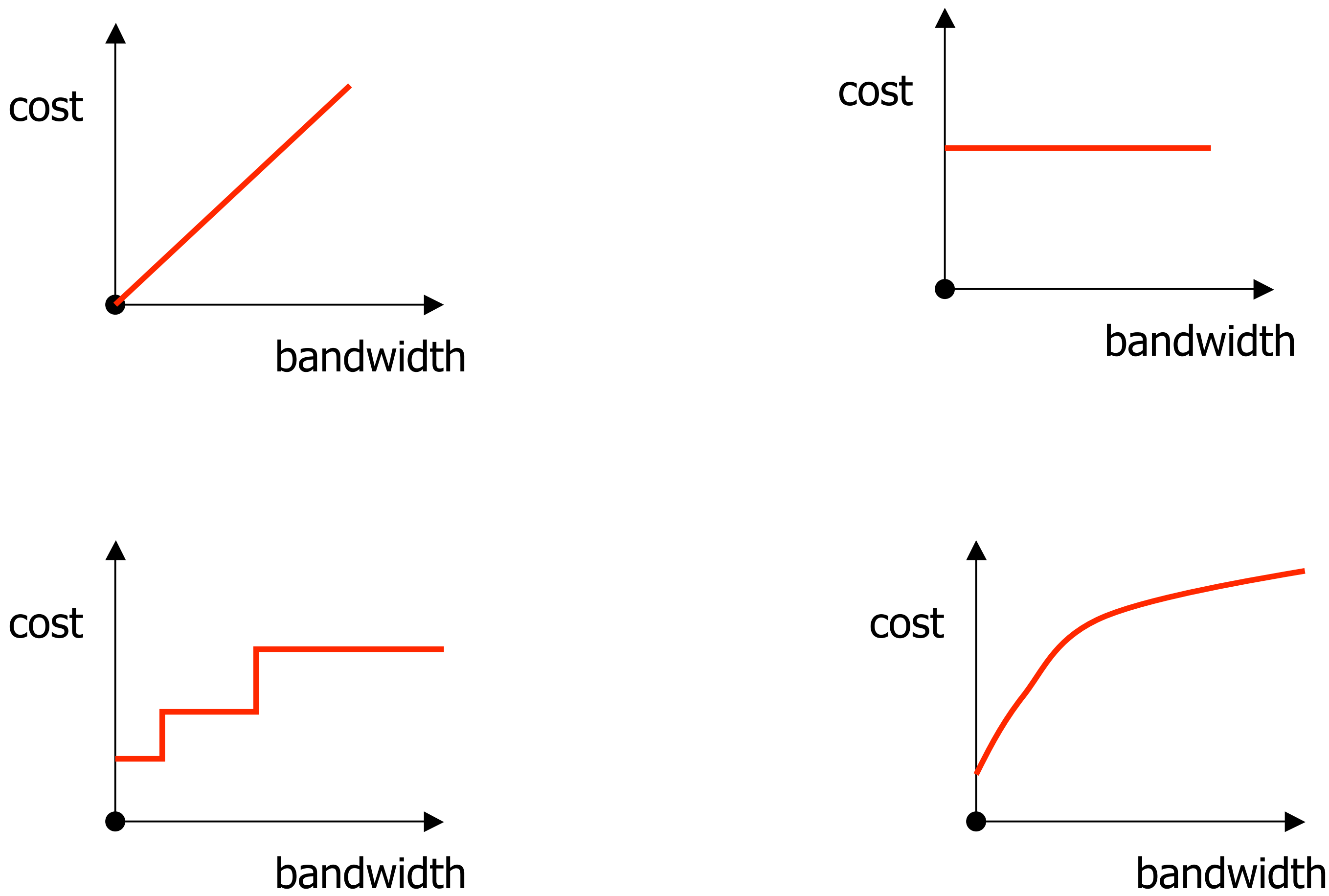
each fiber carries same # of wavelenghts

single-cable model



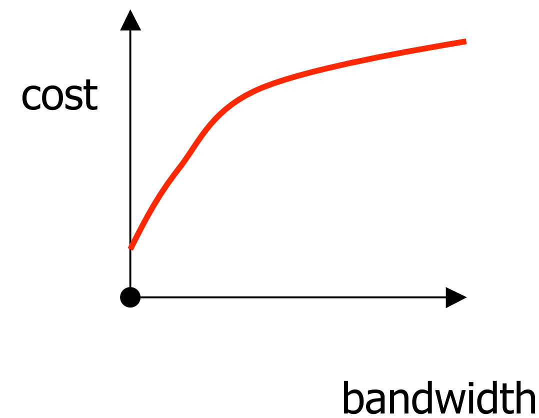
f(x) = minimum # of fibers required for bandwidth of x

Economies of scale: f_e



Sub-additive costs

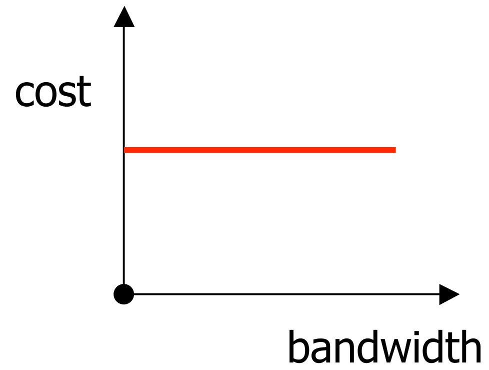
$f_e(x) + f_e(y) \geq f_e(x+y)$



Fixed costs

$f_e(x) = c_e$ for $x > 0$

$= 0$ for $x = 0$



Expresses connectivity

BatB equivalent to Steiner forest problem:

Given $G(V,E)$, $c: E \rightarrow \mathcal{R}^+$ and pairs $s_1t_1, \ldots, s_kt_k$

Find $E' \subseteq E$ s.t for $1 \leq i \leq k$, s_it_i are connected in $G[E']$

minimize $\sum_{e \in E'} c(e)$

NP-hard and APX-hard (best known approx is 2)

Uniform versus Non-uniform

Uniform: $f_e = c_e f$ where $c : E \rightarrow \mathcal{R}^+$

(wlog, $c_e = 1$ for all e, then $f_e = f$)

Non-uniform: f_e different for each edge

Practice:

usually uniform but occasionally non-uniform

Non-uniform problem led to new algorithms and ideas

Heuristic approaches for NP-hard probs

- Integer programming methods
  - branch and bound
  - branch and cut
- approximation algorithms: heuristics guided by analysis and provable guaranteed
- meta-heuristics and ad-hoc methods

Approximation algorithm/ratio

Approximation algorithm $\mathcal{A}$:
polynomial time algorithm

for each instance I,
$\mathcal{A}(I)$ is cost of solution for I given by $\mathcal{A}$
OPT(I) is cost of an optimum solution for I

approximation ratio of $\mathcal{A}$: $\sup_I \mathcal{A}(I)/OPT(I)$

Approximability of Buy at Bulk

	Single-cable	Uniform	Non-Uniform
Single Source (hardness)	O(1) [SCRS'97] Ω(1) folklore	O(1) [GMM'01] Ω(1) folklore	O(log k) [MMP'00] Ω(log log n) [CGNS'05]
Multicommodity (hardness)	O(log n) [AA'97] $\Omega(\log^{1/4-\varepsilon} n)$ [A'04]	O(log n) [AA'97] $\Omega(\log^{1/4-\varepsilon} n)$ [A'04]	$O(\log^4 n)$ [CHKS'06] $\Omega(\log^{1/2-\varepsilon} n)$ [A'04]

Special mention: $2^{(\log n \log\log k)^{1/2}}$ for non-uniform [CK'05]

Three algorithms for multi-commodity

- Using tree embeddings of graphs for *uniform case.* [Awerbuch-Azar'97]
- Greedy routing with randomization and inflation [Charikar-Karagiazova'05]
- Junction based approach [C-Hajiaghayi-Kortsarz-Salavatipour'06]

Alg1: Using tree embeddings

Suppose G is a tree T

Routing is unique/trivial in T

For each $e \in T$, routing induces flow of x_e units

Cost = $\sum_{e \in \mathcal{T}} c_e f(x_e)$

Essentially an optimum solution modulo computing f

Alg1: Using tree embeddings

[Bartal'96,'98, FRT'03]

Given $G=(V, E)$ there is a random tree $T=(V, E_T)$ such that

- $d_T(uv) \geq d_G(uv)$ for each pair uv
- $d_T(uv) \leq O(\log n)\, d_G(uv)$ *in expectation*

(Note: E_T is not related to E)

[AA'97]

Run buy-at-bulk algorithm on T

Claim: Approximation is $O(\log n)$ for *uniform case*

Why only uniform case?

Uniform case: $f_e = c_e \cdot f$ for each e

Tree approximation of G with edge lengths given by c_e

In the non-uniform case, f_e is different for each e, no notion of a metric on V

Open Problems:

- Close gap between $O(\log n)$ upper bound and $\Omega(\log^{1/4-\epsilon} n)$ hardness [Andrews'04]
- Obtain an $O(\log h)$ upper bound where h is the number of pairs

Alg2: Greedy using random permutation

[CK'05]

Assume $d_i = 1$ for all i // *(unit-demand assumption)*

Pick a random permutation of demands

// *(wlog assume $1,2,\ldots,k$ is random permutation)*

for $i = 1$ to k do

  set $d'_i = k/i$ // *(pretend demand is larger)*

  route d'_i for $s_i t_i$ greedily along *shortest path* on cur soln

end for

Details

"route d'_i for s_it_i along *shortest path* on cur soln"

$x_j(e)$: flow on e after j demands have been routed

- compute edge costs $c(e) = f_e(x_{i-1}(e)+1) - f_e(x_{i-1}(e))$ *// additional cost of routing s_it_i on e*
- compute shortest path according to c

Alg2: Theorems

[CK'05]

Theorem: Algorithm is $2^{(\log k \log \log k)^{1/2}}$ approx for non-uniform cost functions

Theorem: Algorithm is $O(\log^2 k)$ approx for uniform cost functions in the single-sink case

Justifies simple greedy algorithm

Key: randomization and inflation

Some empirical evidence of goodness

Alg2: Open Problems

Conjecture: For uniform multi-commodity case, algorithm is polylog(k) approx.

Question: What is the performance of the algorithm in the non-uniform case? polylog(k) ?

Question: Does the natural generalization of the algorithm work (provably) "well" even in the protected case? Not known even for simple connectivity.

Alg3: Junction routing

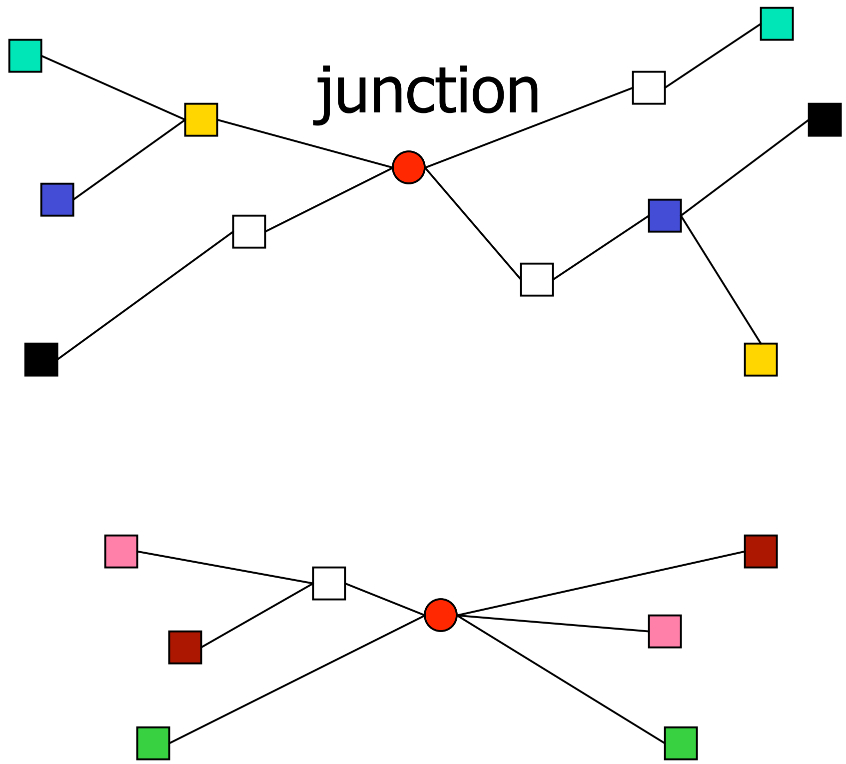
[HKS'05, CHKS'06]

Junction tree routing:

Alg3: Junction routing

[HKS'05, CHKS'06]

Junction tree routing:



Alg3: Junction routing

density of junction tree: cost of tree/# of pairs

Algorithm:

Find a *low density* junction tree T

Remove pairs connected by T

Repeat until no pairs left

Analysis Overview

OPT: cost of optimum solution

Theorem: In any given instance, there is a junction tree of density $O(\log k)\ OPT/k$

Theorem: There is an $O(\log^2 k)$ approximation for a *minimum* density junction tree

Theorem: Algorithm yields $O(\log^4 k)$ approximation for buy-at-bulk network design

Existence of low-density junction trees

Three proofs:

Based on

1. Sparse covers: $O(\log D)$ OPT/k where $D = \sum_i d_i$

2. Spanning tree embeddings: $O(\log^2 k \log \log k)$ OPT/k

3. Probabilistic and recursive partitioning of metric spaces:
   $O(\log k)$ OPT/k

Existence of low-density junction trees

A (weaker) bound of $O(\log^2 k \log \log k)\ OPT/k$

1. Prove that there exists an approximate optimum solution that is a *forest*

2. Use forest structure to show junction tree of good density

Spanning tree embeddings

[Elkin-Emek-Spielman-Teng '05]

Given $G=(V, E)$ there is a probability distribution over *spanning trees* of G such that for a T picked from the distribution, for each pair uv

- $d_T(uv) \geq d_G(uv)$
- $E[d_T(uv)] \leq O(\log^2 n \log\log n)\, d_G(uv)$

Improves previous bound of $2^{(\log n \log\log n)^{1/2}}$ [Alon-Karp-Peleg-West'95]

Forest Solution

Claim: Spanning tree solution implies that there exists an approximate solution to the buy-at-bulk problem s.t

- the edges of the solution induce a *forest*
- the cost of the solution is α OPT where α is the expected distortion bound guaranteed by spanning tree embedding

Reformulation as a two-cost network design problem

Different f_e difficult to deal with.
Simplify problem

each edge e has *two* costs
c_e: fixed cost, need to pay this to use e
l_e: incremental cost, to route flow of x, pay $l_e x$

$f_e(x) = c_e + l_e x$

Above model approximates original costs within factor of 2 [AZ'98,MMP'00]

Objective function

With reformulation, objective function is:

find $E' \subseteq E$ to *minimize*

$$\sum_{e \in E'} c(e) + \sum_{i=1}^{k} d_i \, l_{E'}(s_i, t_i)$$

$l_{E'}$: shortest path distances in $G[E']$

Existence of forest solution

$E^* \subseteq E$ an optimum soln, $G^* = G[E^*]$

Apply [EEST'05] to G^* with edge lengths l

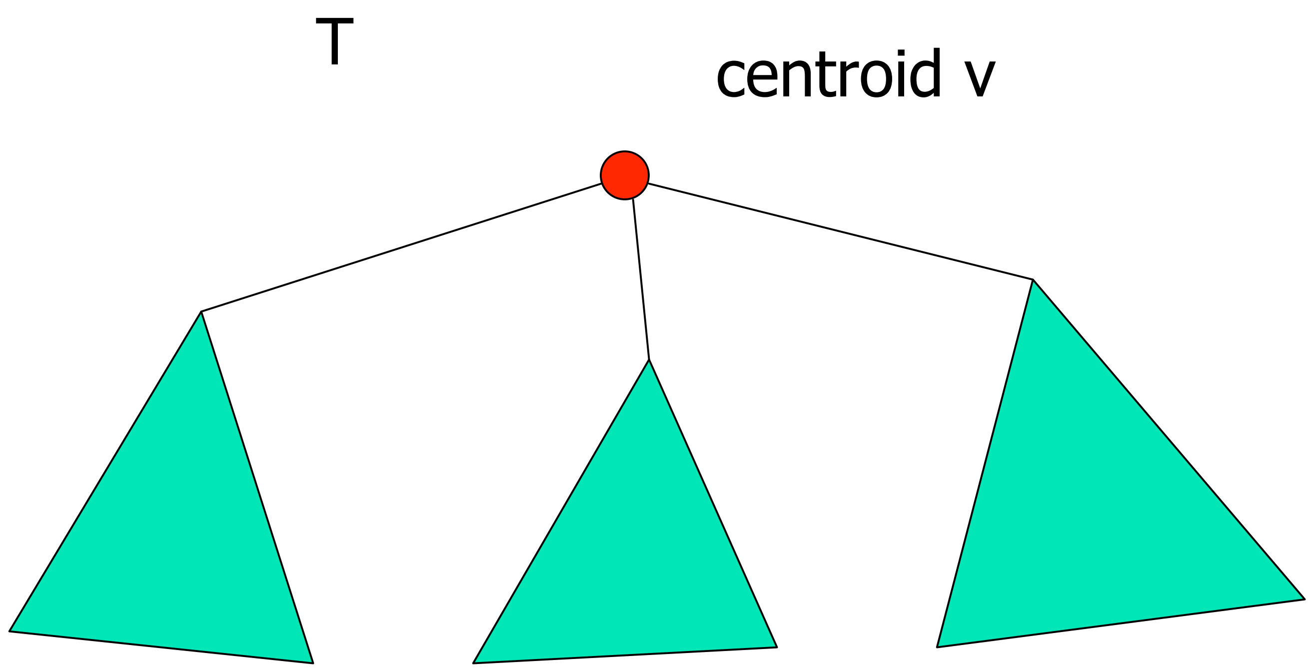
There exists spanning tree T of G^* s.t

$l_T(uv) = O(\log^2 n \log\log n)\, l_{E^*}(uv)$ in expectation

therefore

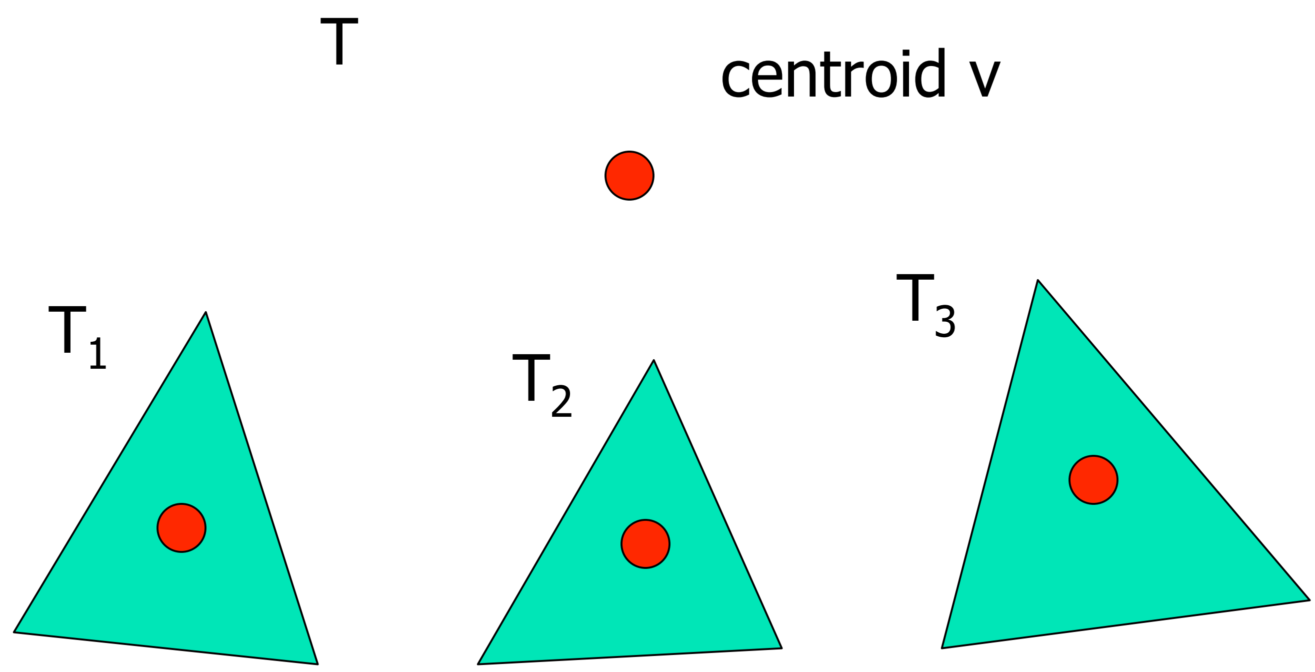
$c(E(T))+\sum_i l_{E(T)}(s_i t_i) \leq c(E^*) + O(\log^2 n \log\log n) \sum_i l_{E^*}(s_i t_i)$

Forest solution to junction tree

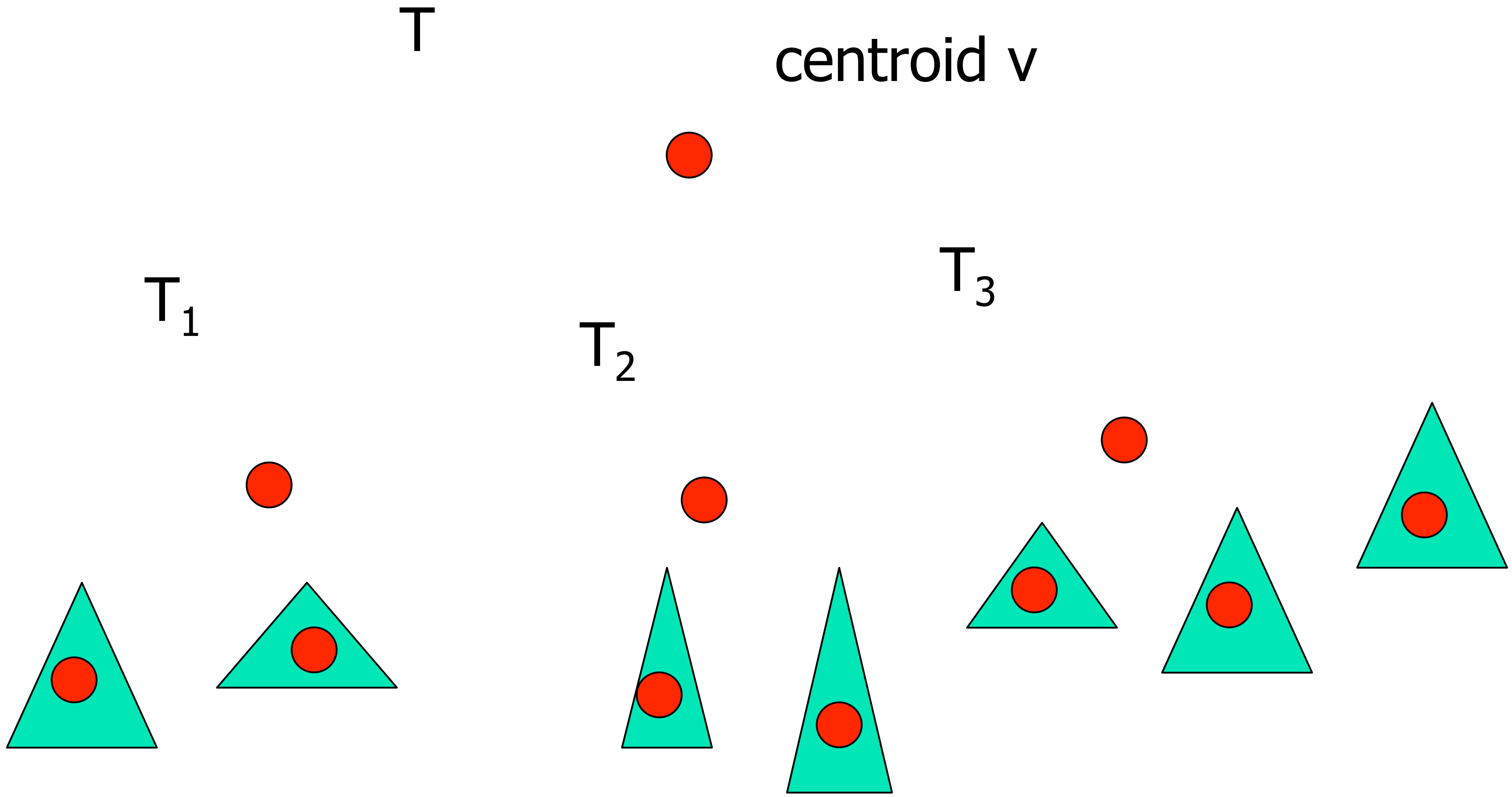


If k/log k terminals have lca = v, done

Forest solution to junction tree



Forest solution to junction tree



Claim: one of these junction trees has density O(log k) den(T)

Finding low-density junction trees

Closely related to single-source buy-at-bulk prob.

Single source problem:

source s, terminals $t_1, t_2, \ldots, t_k$

demand d_i from s to t_i

Goal: route all pairs to minimize cost

Min-density problem for single source:

Goal: connect subset of pairs to minimize

density = cost/# of pairs connected

Single-source BatB

Single source problem:

source s, terminals t_1, t_2, ..., t_k

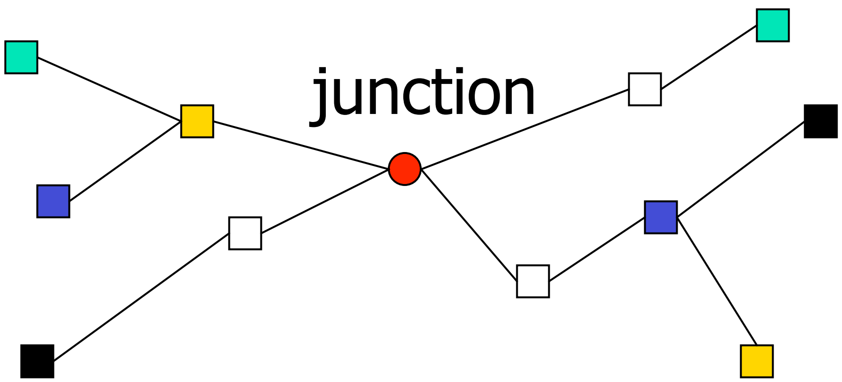
demand d_i from s to t_i

Goal: route all pairs to minimize cost

[Meyerson-Munagala-Plotkin'00] An $O(\log k)$ randomized combinatorial approx.

[C-Khanna-Naor'01] A deterministic $O(\log k)$ approx and integrality gap for natural LP

Min-density junction tree



Similar to single-source? Assume we know junction r.

Two issues:

- which pairs to connect via r?
- how do we ensure that both s_i and t_i are connected to r?

Min-density junction tree

[CHKS'06]

Theorem: α approx for single-source via natural LP implies an $O(\alpha \log k)$ approx for min-density junction tree

Using [CKN'01], $O(\log^2 k)$ approx for min-density junction tree

Approach is generic and applies to other problems as well

Alg3: Open Problems

- Close gap for non-uniform: $\Omega(\log^{1/2-\varepsilon} n)$ vs $O(\log^4 n)$
  - [Kortsarz-Nutov'07] improve to $O(\log^3 n)$ for polynomial demands
  - LP integrality gap?

- Tight bounds for embedding into spanning trees. [EEST'05] show $O(\log^2 n \log\log n)$ and lower bound is $\Omega(\log n)$. Planar graphs?

Buy-at-Bulk with Protection

For each pair $s_i t_i$ send data simultaneously on two *node disjoint paths* P_i (primary) and Q_i (backup)

Protection against equipment failures

Easier case: P_i and Q_i are edge disjoint

Related to Steiner network problem (survivable network design problem)

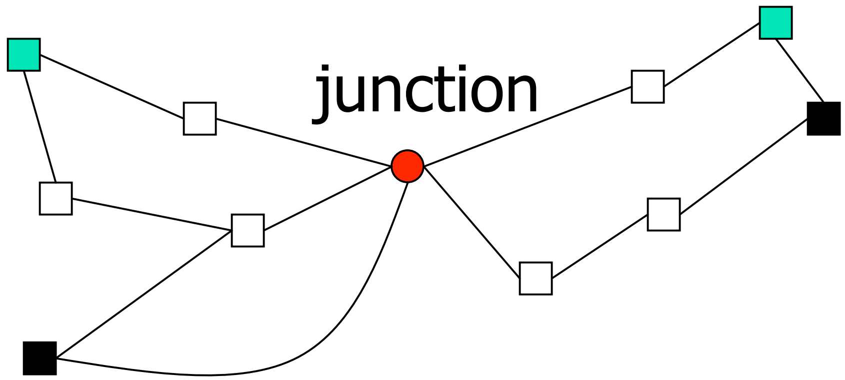
[Jain'00, Fleischer-Jain-Williamson'04]

Buy-at-Bulk with Protection

Junction scheme?

Edge disjoint case easier

2-edge-disj paths from s_i to junction *and* 2-edge-disj-paths from t_i to junction

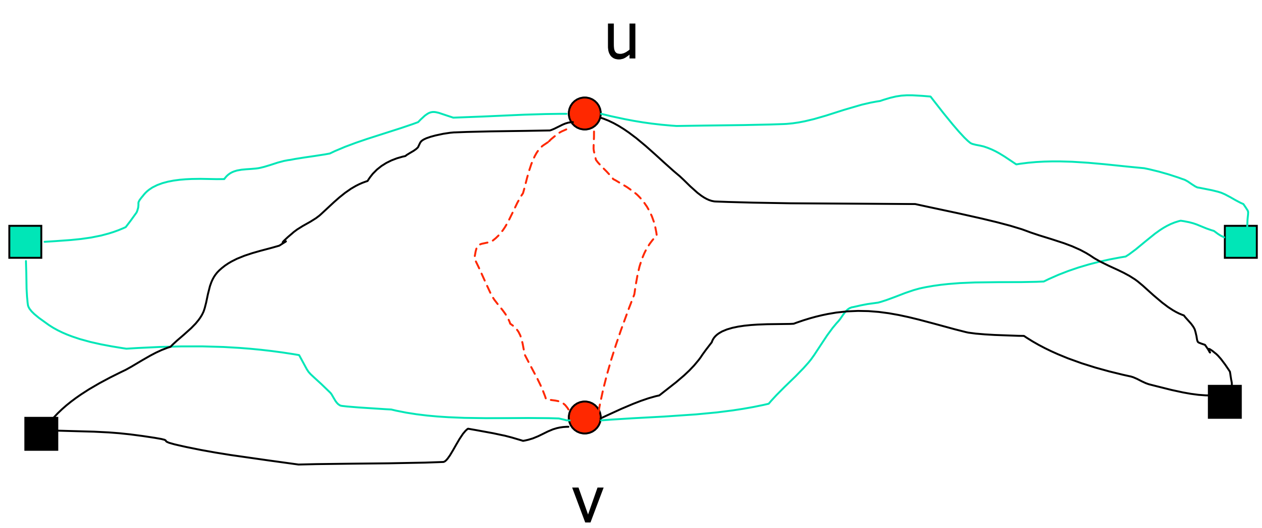


Buy-at-Bulk with Protection

Node disjoint case:

[Antonakopoulos-C-Shepherd-Zhang’07]

2-junction scheme:



Buy-at-Bulk with Protection

[ACSZ'07]

2-junction-Theorem: α-approx for single-source problem via natural LP implies $O(\alpha \log^3 h)$ for multi-commodity problem

Technical challenges

- junction density proof (only one of the proofs in three can be generalized with some work)
- single-source problem not easy! $O(1)$ for single-cable [ACSZ'07]

Open Problems: Single-source for uniform and non-uniform

Conclusion

- Buy-at-bulk network design useful in practice *and* led to several new theoretical ideas
- Algorithmic ideas:
  - application of Bartal's tree embedding [AA'97]
  - derandomization and alternative proof of tree embeddings [CCGG'98,CCGGP'98]
  - hierarchical clustering for single-source problems [GMM'00,MMP'00,GMM'01]
  - cost sharing, boosted sampling [GKRP'03]
  - junction scheme [CHKS'06]
- Hardness of approximation:
  - canonical paths/girth ideas for routing problems [A'04]
- Several open problems

Routing in Practice

Joint with S. Antonakopoulos and S. Fortune

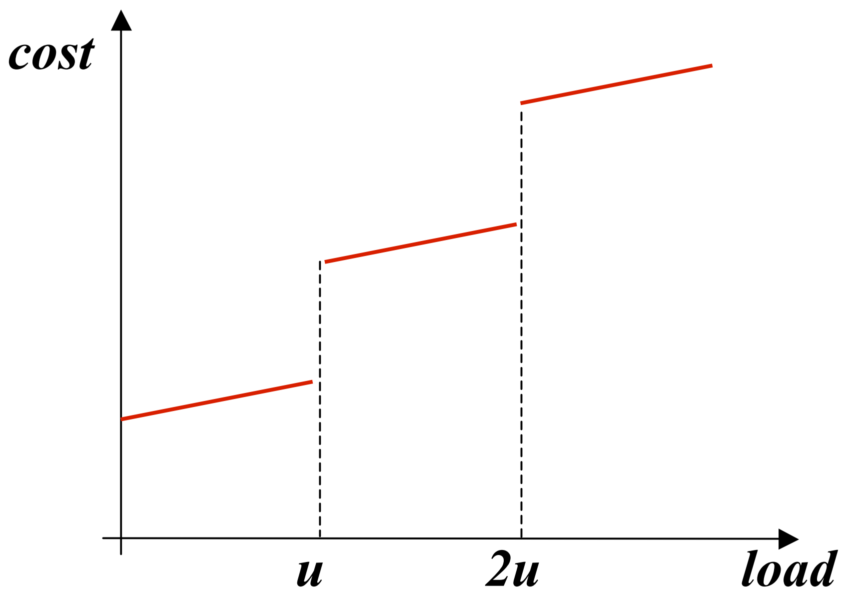
Simple, flexible and scalable heuristics

- Accommodate messy and ever changing requirements
  - Some links may have hard capacity
  - Some nodes may have degree bound
  - Some demands may have forbidden links/nodes
  - Different fiber types, different protection specification
  - Dual homing, multicast…
- Accommodate problem instances of varying sizes
- Close to optimality
  - Typical network costs hundreds of million dollars
  - Small percentage error desired
  - Optimal solution for small/test instances
- Cannot rely on commercial solvers/tools

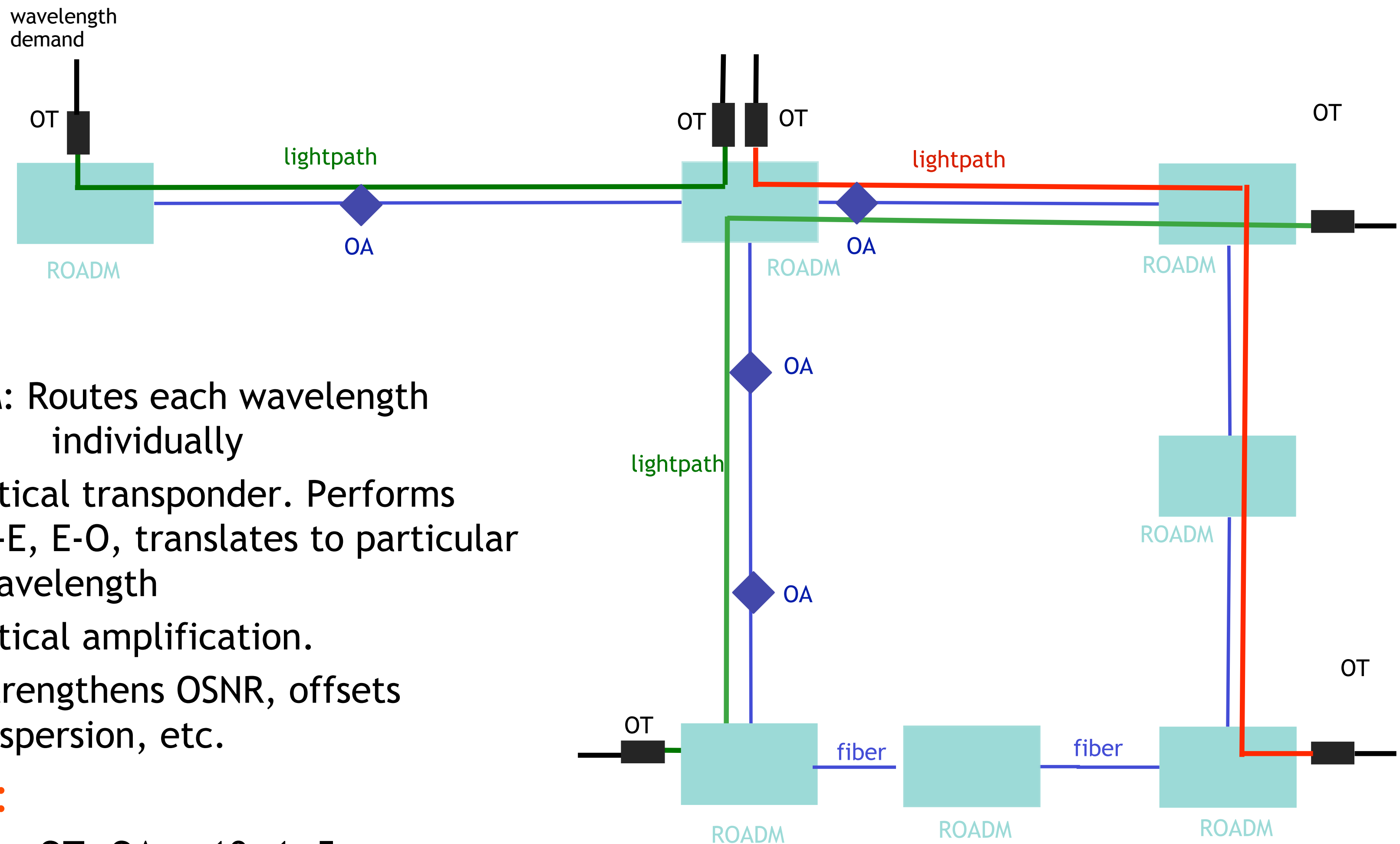
Modeling cost

Cost $f_e(w)$ of a WDM fiber on edge e

- $f_e(w) = c_1 * \lceil w/u \rceil + c_2 * l * \lceil w/u \rceil + c_3 * l * w$
- w: current load, l : length of e, u: fiber capacity
- c_1, c_2, c_3: parameters defined by equipment properties



Optical Components



ROADM: Routes each wavelength individually

OT: optical transponder. Performs O-E, E-O, translates to particular wavelength

OA: optical amplification.
Strengthens OSNR, offsets dispersion, etc.

Costs:

ROADM : OT: OA ~ 10: 1: 5

Modeling cost

$$f_e(w) = c_1 * \lceil w/u \rceil + c_2 * l * \lceil w/u \rceil + c_3 * l * w$$

- $\lceil w/u \rceil$ fibers over e
- One arm of ROADM connects to one end of a fiber :

  c_1 = 2 * cost(1-arm ROADM)
- Each OA amplifiers signals (per fiber basis), over distance reach(OA) :

  c_2 = cost(OA) / reach(OA)
- Each OT converts signal O-E or E-O (per wavelength basis), over distance reach(OT):

  c_3 = cost(OT) / reach(OT)

Basic greedy algorithm

Process each demand in turn

- For each edge, calculate the marginal cost of routing the demand through the edge

$$f_e(w + d) - f_e(w)$$

- Calculate shortest disjoint paths using marginal costs as weights.
- Route the demand via these paths.

Theoretical link: [Charikar-Karagiazova'05]

Improvements

Ordering of processing is critical

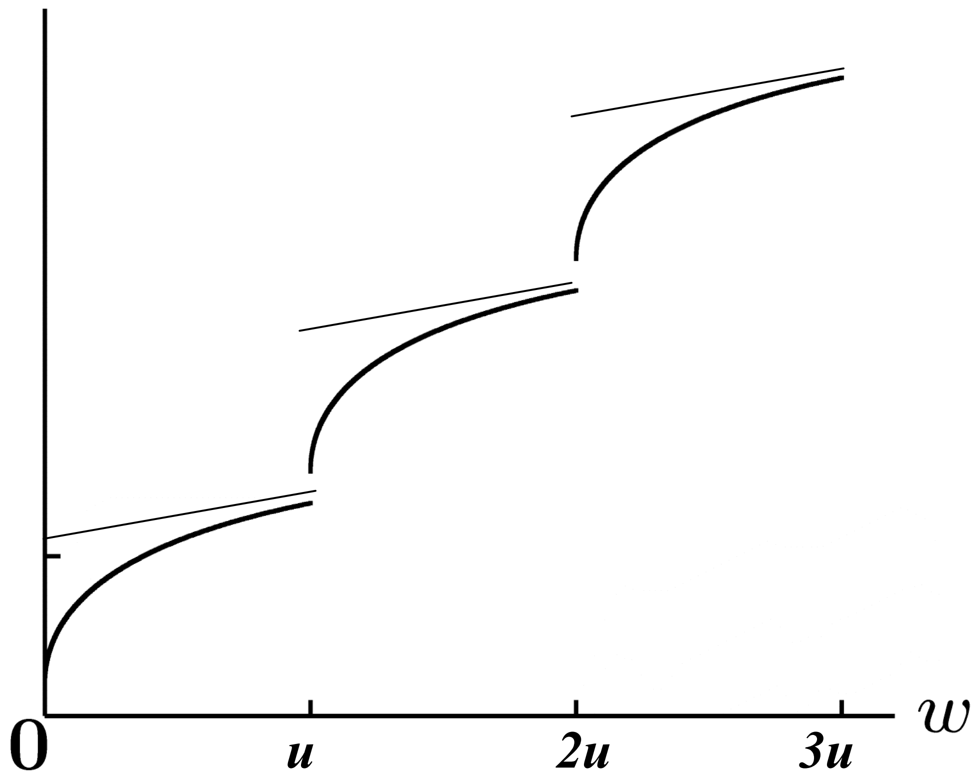
- No simple a priori criterion that defines an "optimal" order.
- Best solution usually obtained by trying several random orderings.

Iterative refinement: Process each demand again to find shortest paths in then-current network

- Converges monotonically to a local optimum, typically in less than 10 passes.
- Very large and/or heavily loaded instances may require more passes.

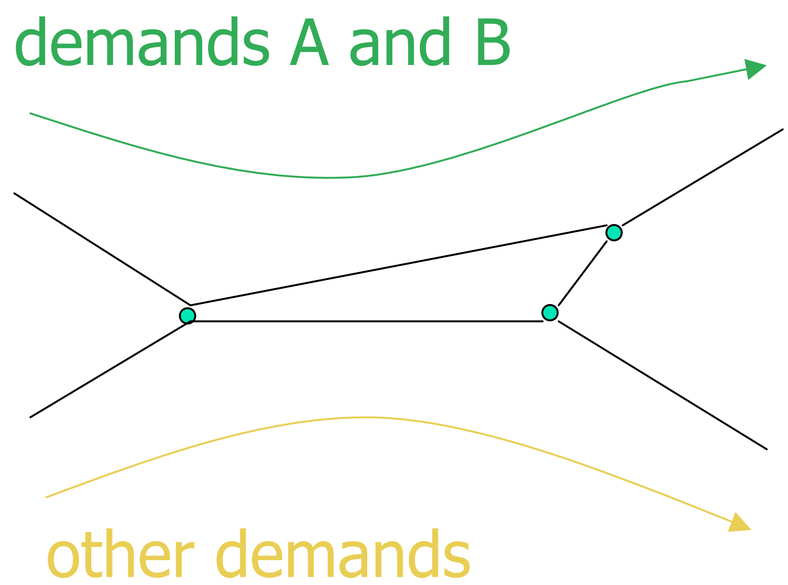
Improvements (cont)

Calculate marginal costs using a piecewise strongly concave pseudocost function.



Example

Advantage: free lightly loaded fibers for cost reduction



Handling extra constraints

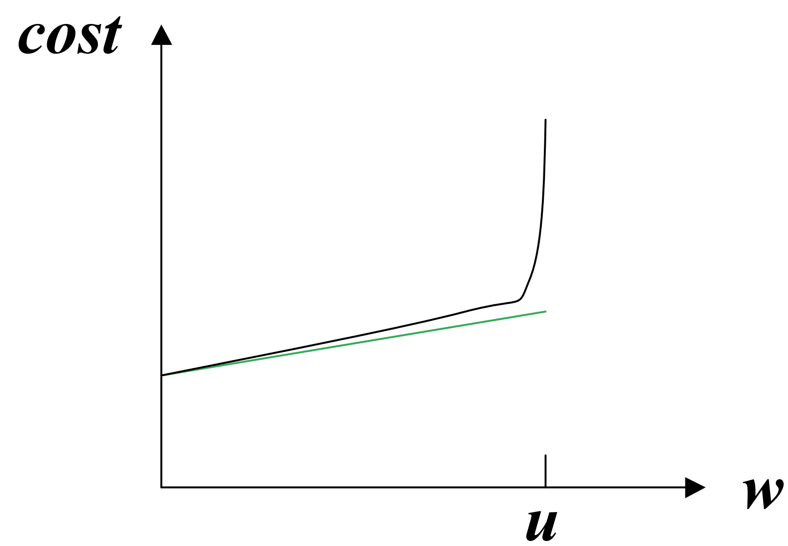
Example: capacitated edges

- Primary obj: route as many demands as possible
- Secondary obj: cost minimization

Penalty heuristic

“Penalize” demands that use almost-full edges.

- In subsequent iterations, some capacity in highly loaded edges freed up. More demands may be routed.

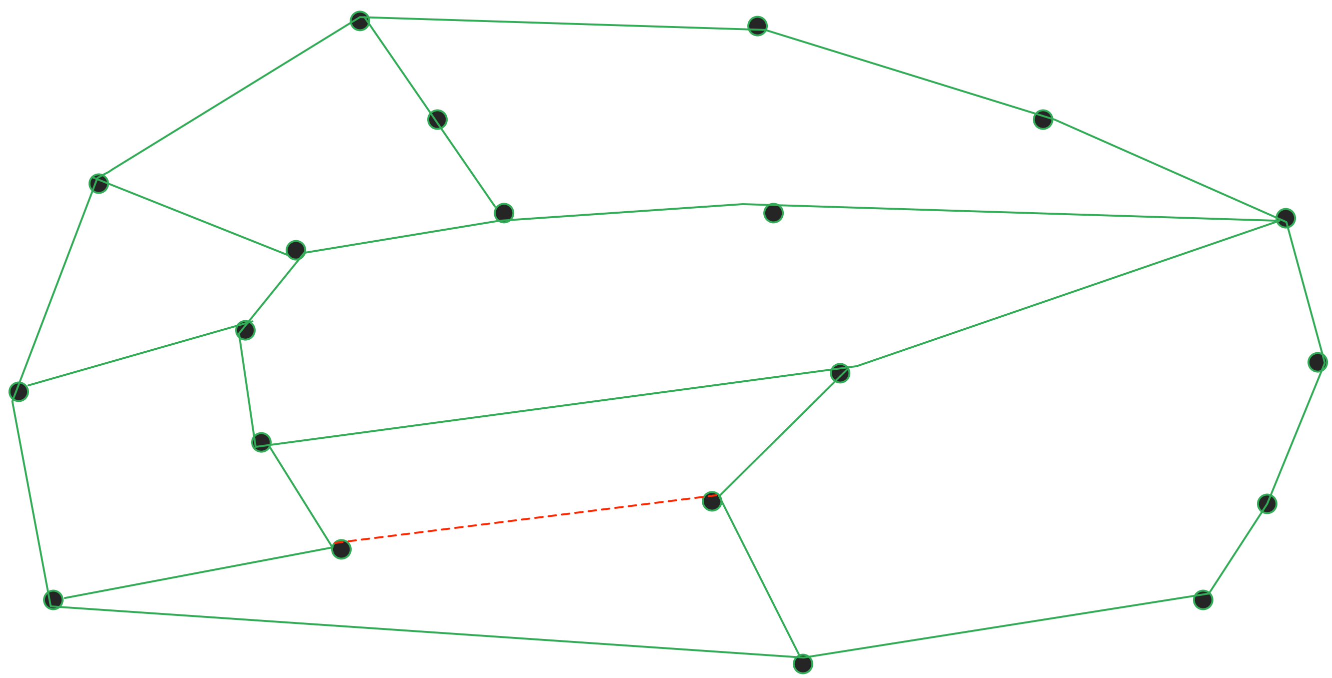


Penalty heuristic (contd.)

- Harshness of the penalty is adaptive, depending on the percentage of unroutable demands.
- Converges monotonically w.r.t. the number of unroutable demands (but not cost).
- If all demands are successfully routed, may switch to reducing cost, by additional iterative refinement and pseudocost.

Example

- Without penalty function, many demands cannot be routed.
- Fewer unrouted demands when red link removed, somewhat unexpectedly.



Example (cont)

- With penalty function, all demands routed.
- Higher probability that a random demand ordering will yield a good solution when edge is missing!
- Optimal solution noticeably worse with red edge missing, as expected.
- Best solutions found by the heuristic within 1% of respective optima.

Performance

	Heuristic		*Optimum*	
Instance	*Cost*	*Unrouted*	*Cost*	*Unrouted*
A	5220172	0	5167850	0
B	5371217	0	5362191	0
C	10831734	0	10133087	0

	Heuristic		*Optimum*	
Instance	*Cost*	*Fibers*	*Cost*	*Fibers*
D	103525	25	102592	24
E	131237	26	131237	26
F	89623	25	89565	26
G	113759	25	113697	25
H	135871	27	135863	27

Wavelength Assignment

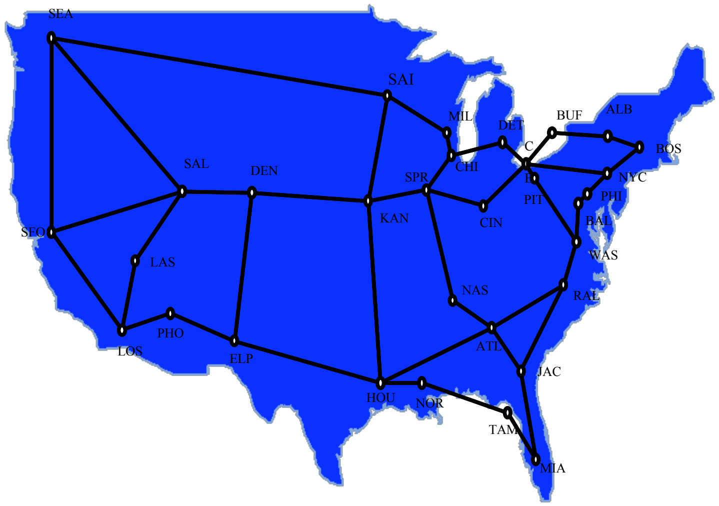
Design Problem

Input

- A network
- Demands

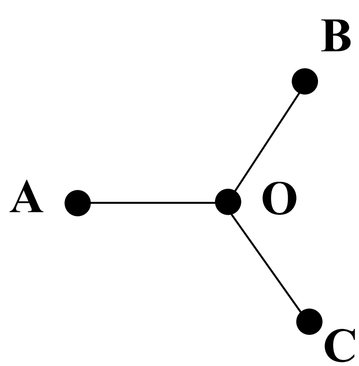
Output for each demand

- Routing
- Wavelength assignment

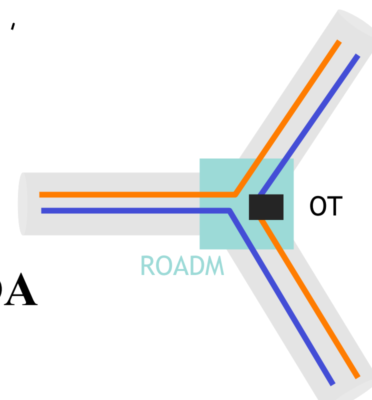


Wavelength assignment

- Demand paths sharing same fiber have distinct wavelengths
- Deploy no extra fibers
- Use convertors (OT) if necessary
- Min number of convertors



Fiber capacity u = 2
Demand routes:
AOB, BOC, COA

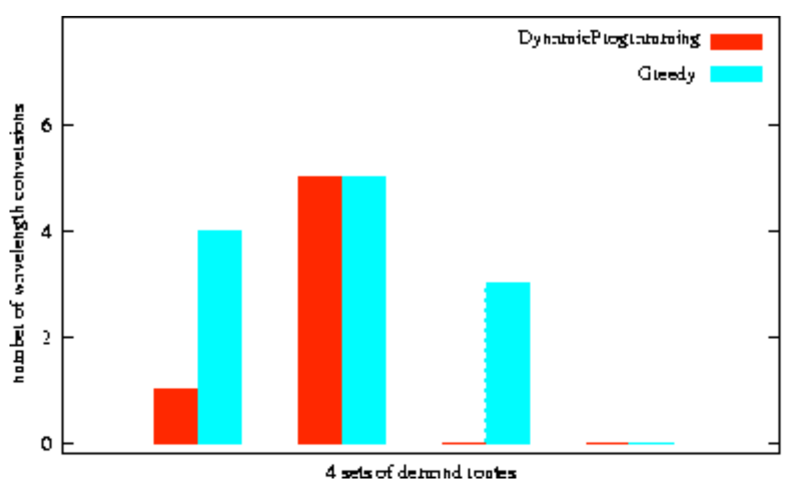


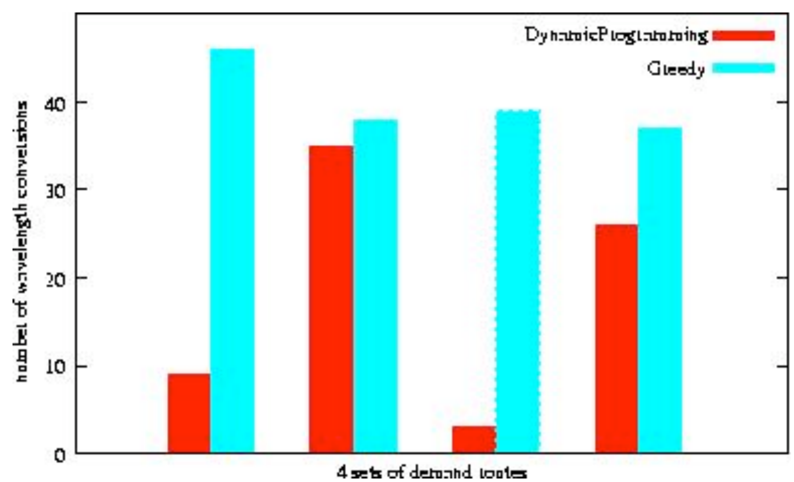
Heuristics

- Limited theoretical results
- Practical heuristics
  - Dynamic programming:
    - Routing path for demand d : e_1, e_2, …
    - C(e_i, λ, f) : min number of conversions needed for subpath e_1, … e_i if e_i is assigned wavelength λ on fiber f
    - $C(e_i, \lambda, f) = \min\{ \min_g C(e_{i-1}, \lambda, g) , \min_{\lambda' \neq \lambda, g} C(e_{i-1}, \lambda', g) + 1 \}$
  - Greedy approach
    - On link e_i, continue with same wavelength λ if possible or switch to λ' that is feasible on the most number of subsequent links
  - Trade off in performance and running time

Heuristics

- Trade off in performance and running time

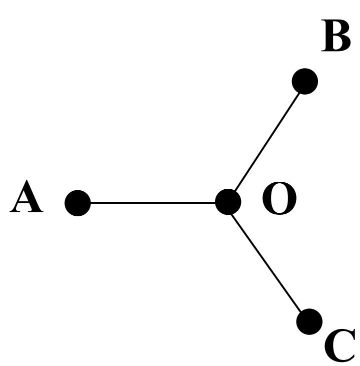




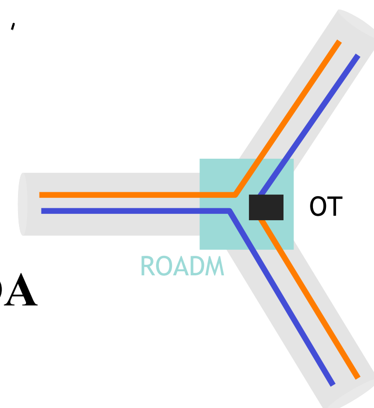
Wavelength assignment

Model 1: min conversion

- Demand paths sharing same fiber have distinct wavelengths
- Deploy no extra fibers
- Use convertors (OT) if necessary
- Min number of convertors



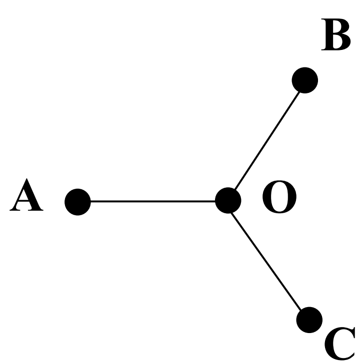
Fiber capacity u = 2
Demand routes:
AOB, BOC, COA



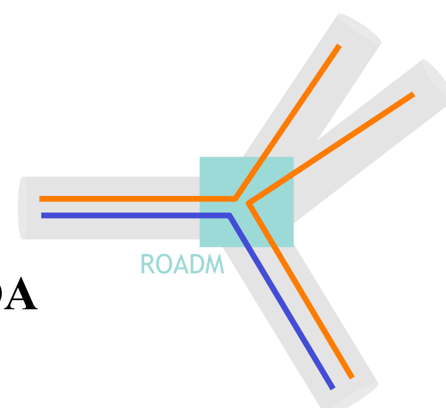
Wavelength assignment

Model 2: min fiber without conversion

- Each demand path assigned one wavelength from src to dest – no conversion
- Demand paths sharing common fiber have distinct wavelengths
- Deploy extra fibers if necessary
- Min total fibers



Fiber capacity u = 2
Demand routes:
AOB, BOC, COA



Results

Network is a line (WinklerZ)

- Optimally solvable
- $f(e)$ fibers necessary and sufficient on every link e
- u : fiber capacity
- $w(e)$: load on link e
- $f(e) = \lceil w(e) / u \rceil$

Tree (ChekuriMydlarzShepherd)

- NP hard
- 4 approx for trees: $4 f(e)$ fibers sufficient on e

Results (cont)

Hard to approx for arbitrary topologies (AndrewsZ)

Inapprox ratio	*Total fiber*	*Max fiber per edge*
Routing + WA	$(\log M)^{1/4-\varepsilon}$	$(\log\log M)^{½-\varepsilon}$
WA (given routing)	Any constant	$(\log u)^{½-\varepsilon}$

Results (cont)

Hard to approx for arbitrary topologies (AndrewsZ)

Inapprox ratio	*Total fiber*	*Max fiber per edge*
Routing + WA	$(\log M)^{1/4 - \varepsilon}$	$(\log \log M)^{½- \varepsilon}$
WA (given routing)	Any constant	$(\log u)^{½- \varepsilon}$

Buy-at-bulk

Congestion minimization

Chromatic number

3SAT(5), Raz verifier

Results (cont)

Hard to approx for arbitrary topologies (AndrewsZ)

Inapprox ratio	*Total fiber*	*Max fiber per edge*
Routing + WA	$(\log M)^{1/4 - \varepsilon}$	$(\log \log M)^{½- \varepsilon}$
WA (given routing)	Any constant	$(\log u)^{½- \varepsilon}$

Logarithmic approx for arbitrary topologies

Approx ratio	*Total fiber*	*Max fiber per edge*
Routing + WA	$O(\log M)$	$O(\log M)$
WA (given routing)	$O(\log u)$	$O(\log u)$

Heuristics

Greedy approach: For each demand choose a wavelength that increases fiber count least

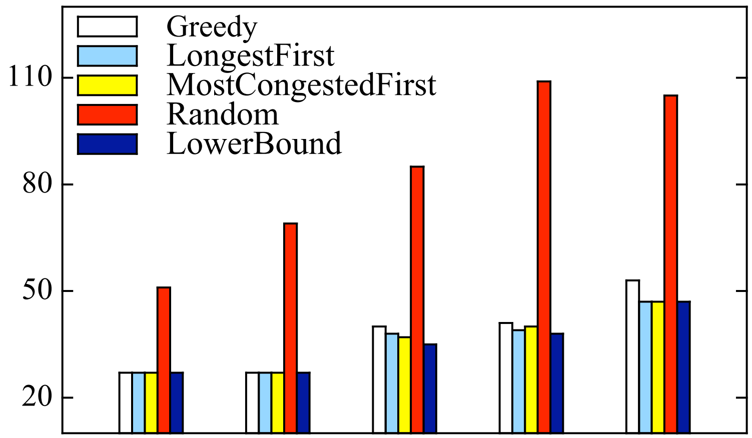
1. Basic greedy: demands handled in a fixed given order
2. Longest first: demands with more hops first
3. Most congested first: demands with congested routes first

Randomized assignment

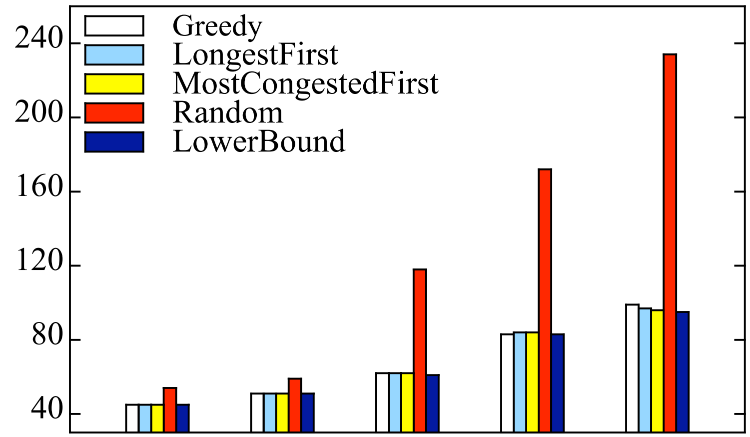
- Choose a wavelength [1, u] uniformly at random for each demand;
- O(log u) approx

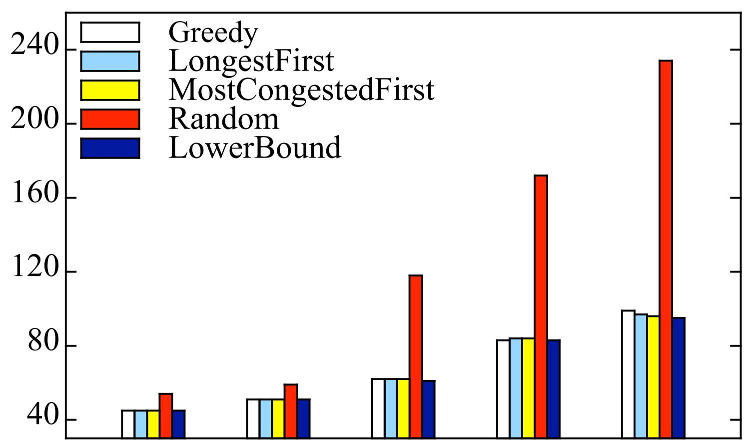
Optimal solution via integer programming

Performance on 3 US backhaul networks



Network A: 5-year traffic



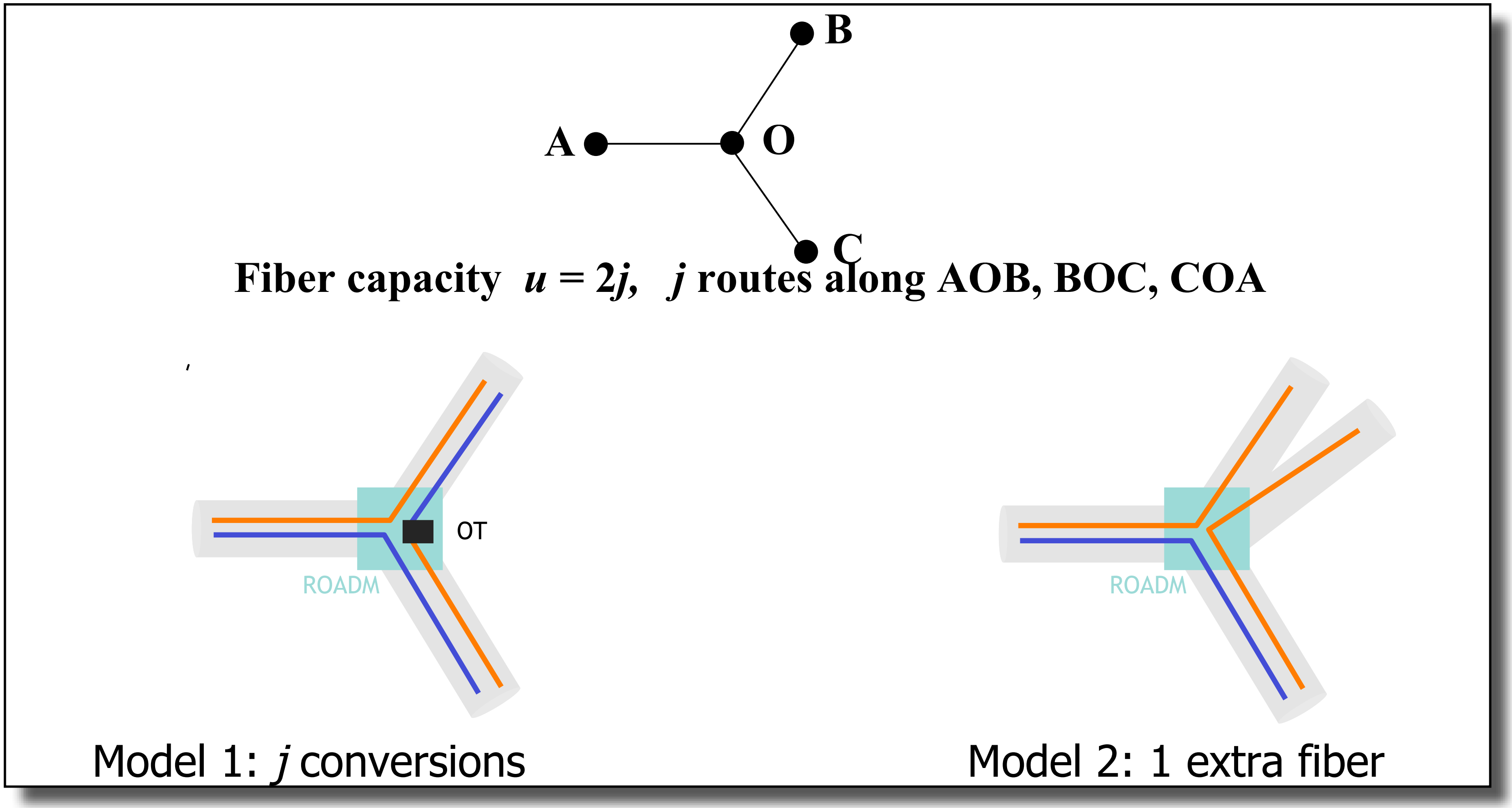


Why not randomization?

- Birthday paradox:
  If load >√ u, some wavelength chosen twice with prob > ½
- If load = u, some wavelength chosen log u time whp.

Open issue: Model 1 vs model 2

- Two models studied in isolation
- Which is more cost effective?



Conclusion

- Optical network design extremely complex
- Smaller pieces hard to optimize
  - Routing: buy-at-bulk network design
  - Wavelength assignment
  - Physical layer optimization
- Gap between theoretical knowledge and practical implementability