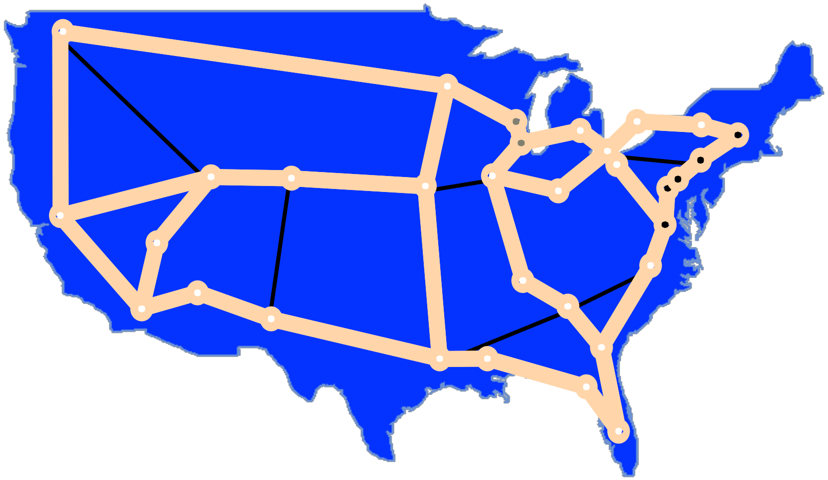
Buy at Bulk Network Design (with Protection)

Chandra Chekuri
Univ. of Illinois, Urbana-Champaign

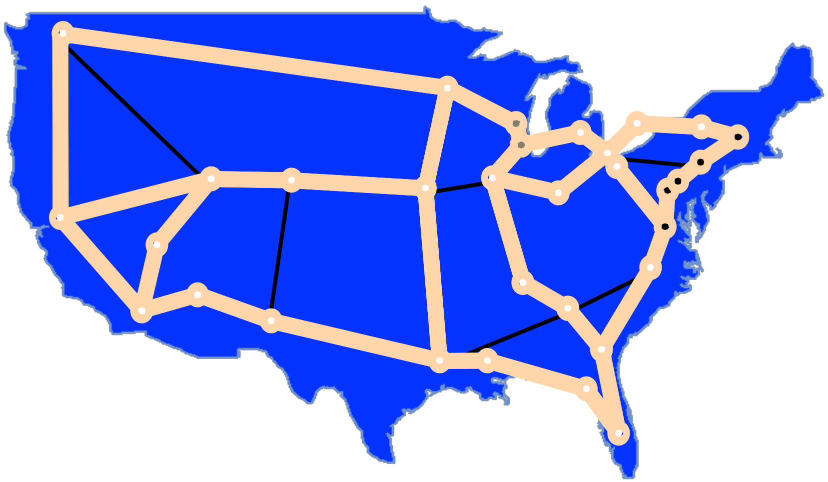
Optical Network Design



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

Objectives: minimize cost, maximize fault tolerance, …

Optical Network Design



Details: see tutorial talk by C-Zhang, DIMACS workshop on Next Gen Networks, August 2007

Buy-at-Bulk Network Design

[Salman-Cheriyan-Ravi-Subramanian'97]

Network: graph $G=(V,E)$

Cost functions: for each $e \in E$, $f_e: \mathbb{R}_+ \to \mathbb{R}_+$

Demand pairs: $s_1t_1, s_2t_2, \ldots, s_ht_h$ (multicommodity)

Demands: s_it_i has a positive demand d_i

Buy-at-Bulk Network Design

Feasible solution:

- a multi-commodity flow for the given pairs
- d_i flow from s_i to t_i (can also insist on unsplittable flow along a single path)

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

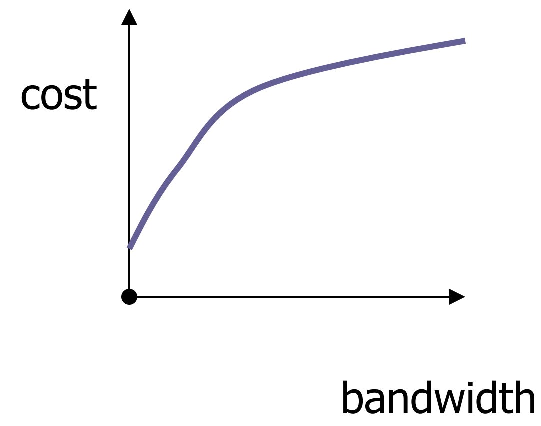
Goal: minimize cost of flow

Single-sink BatB

Sink s, **terminals** t_1, t_2, …, t_h , **demand** d_i from t_i to s

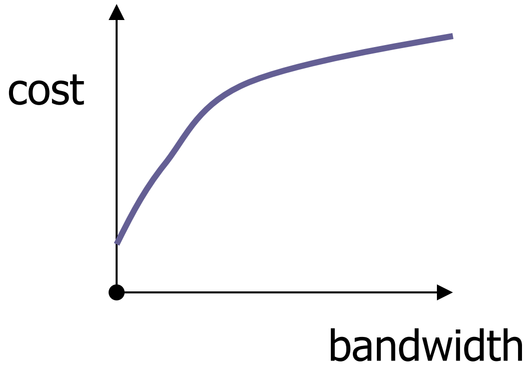
Economies of scale (sub-additive costs)

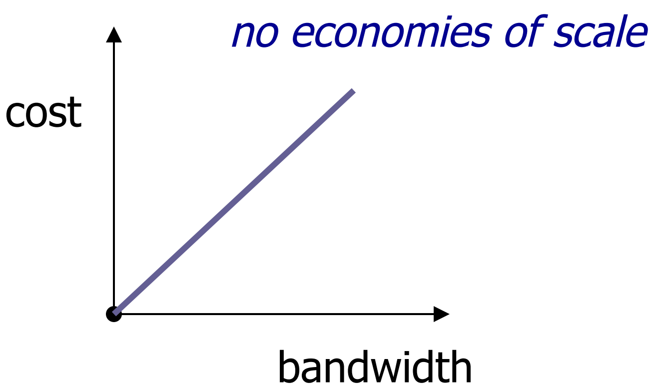
$f_e(x) + f_e(y) \geq f_e(x+y)$

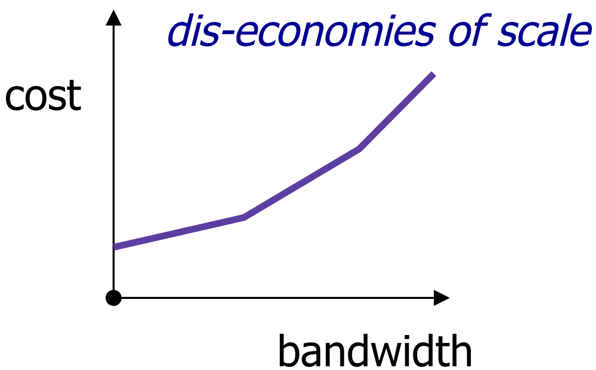


Economies of scale (sub-additive costs)

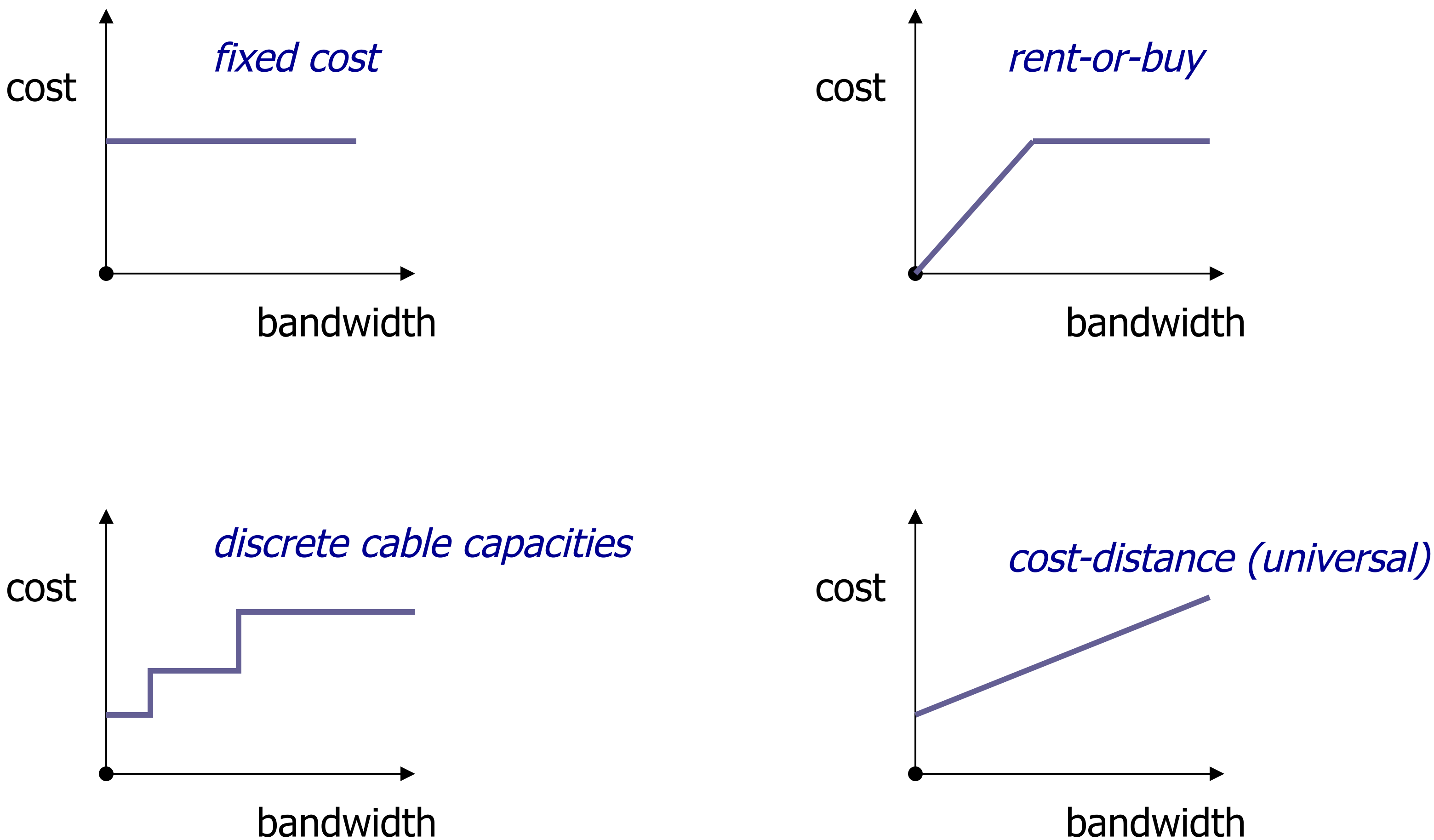
$f_e(x) + f_e(y) \geq f_e(x+y)$







Economies of scale



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

(can assume wlog is a simple cost-distance function)

Throughout talk graphs are *undirected*

Approximability

	Single-cable	Uniform	Non-Uniform
Single Source (hardness)	$O(1)$ [SCRS' 97] $\Omega(1)$ folklore	$O(1)$, 20.42 [GMM'01, GR'10] $\Omega(1)$ folklore	$O(\log h)$ [MMP' 00] $\Omega(\log \log n)$ [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]	$O(\log^4 h)$* [CHKS' 06] $\Omega(\log^{1/2-\varepsilon} n)$ [A' 04]

*$O(\log^3 n)$ for poly-bounded demands [KN'07]

Easy to state open problems

- Close gaps in the table
- Improved bounds for planar graphs or geometric instances?

Three algorithms for Multi-commodity BatB

- 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 = $\sum_e c_e f(x_e)$

Essentially an optimum solution modulo computing f

Alg1: Using tree embeddings

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

Theorem: O(log n) distortion for embedding a n point finite metric into random dominating tree metrics

[Awerbuch-Azar'97]

Theorem: O(log n) approximation for multicommodity buy-at-bulk with *uniform cost functions*

Open problems for uniform

- 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~~ *follows from refinement of tree embeddings due to* [Gupta-Viswanath-Ravi'10]

Alg2: Greedy using random permutation

[Charikar-Karagiozova'05] (inspired by [GKRP'03] for rent-or-buy)

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

Pick a random permutation of demands

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

for $i = 1$ to h do

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

route d'_i for $s_i t_i$ greedily along *shortest path* on current solution

end for

Details

"route d'_i for $s_i t_i$ along *shortest path* on current solution"

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

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

Alg2: Theorems

[CK'05]

Theorem: Algorithm is $2^{O(\sqrt{\log h \log \log h})}$ approx. for *non-uniform* cost functions.

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

- Justifies simple greedy algorithm
- Key: randomization and inflation
- Some empirical evidence of goodness

Alg2: Open Problems

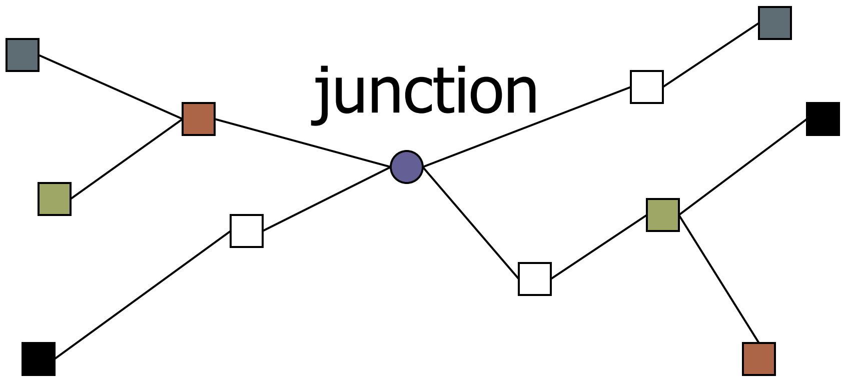
Question/Conjecture: For uniform multi-commodity case, algorithm is polylog(h) approx.

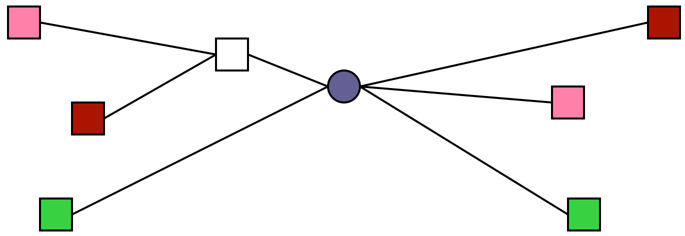
Question: What is the performance of the algorithm in the non-uniform case? polylog(h) ?

Alg3: Junction routing

[HKS'05, CHKS'06]

Junction tree routing:





Alg3: Junction routing

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

Algorithm:

While demand pairs left to connect *do*

- Find a *low density* junction tree T
- Remove pairs connected by T

Analysis overview

OPT: cost of optimum solution

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

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

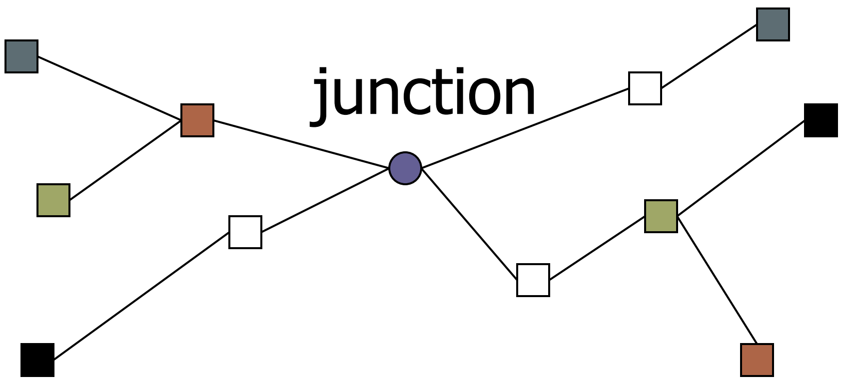
Theorem: Algorithm yields $O(\log^4 h)$ approximation for buy-at-bulk network design

Existence of good junction trees

Three proofs:

1. Sparse covers: $O(\log D)$ OPT/h where $D = \sum_i d_i$
2. Spanning tree embeddings: $\tilde{O}(\log h)$ OPT/h
3. Probabilistic and recursive partitioning of metric spaces: $O(\log h)$ OPT/h

Min-density junction tree



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

Two issues:

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

Min-density junction tree

[CHKS'06]

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

Via [C-Khanna-Naor'01] on single-source LP gap, $O(\log^2 h)$ approximation.

Approach is generic and applies to other problems

Alg3: Open Problems

Close gap for non-uniform: $\Omega(\log^{1/2-\varepsilon} n)$ vs $O(\log^4 h)$

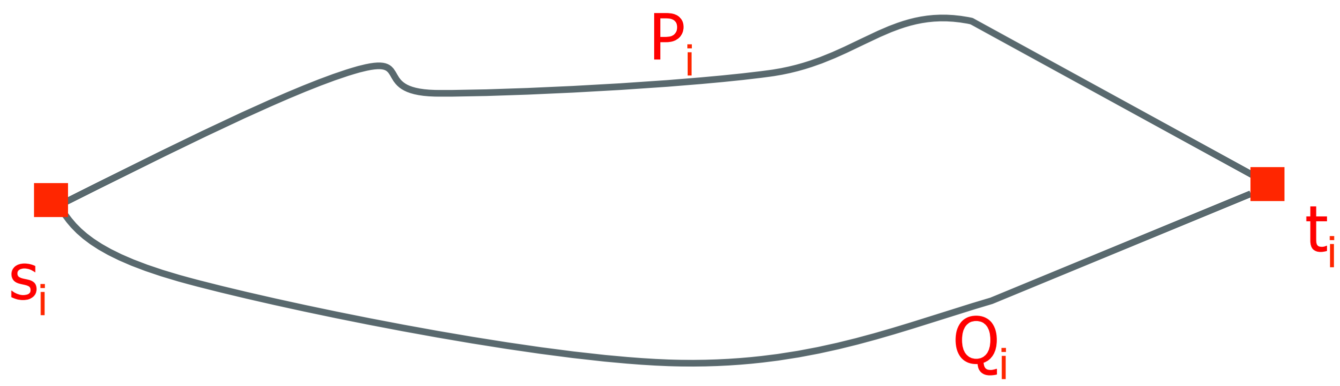
- [Kortsarz-Nutov'07] improved to $O(\log^3 n)$ for polynomial demands
- Junction tree analysis is with respect to integral solution. What is the integrality gap of the natural LP?

Buy-at-Bulk with Protection

(1+1)-protection in practical optical networks

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

Protection against equipment/link failures



Buy-at-Bulk with Protection

More generally:

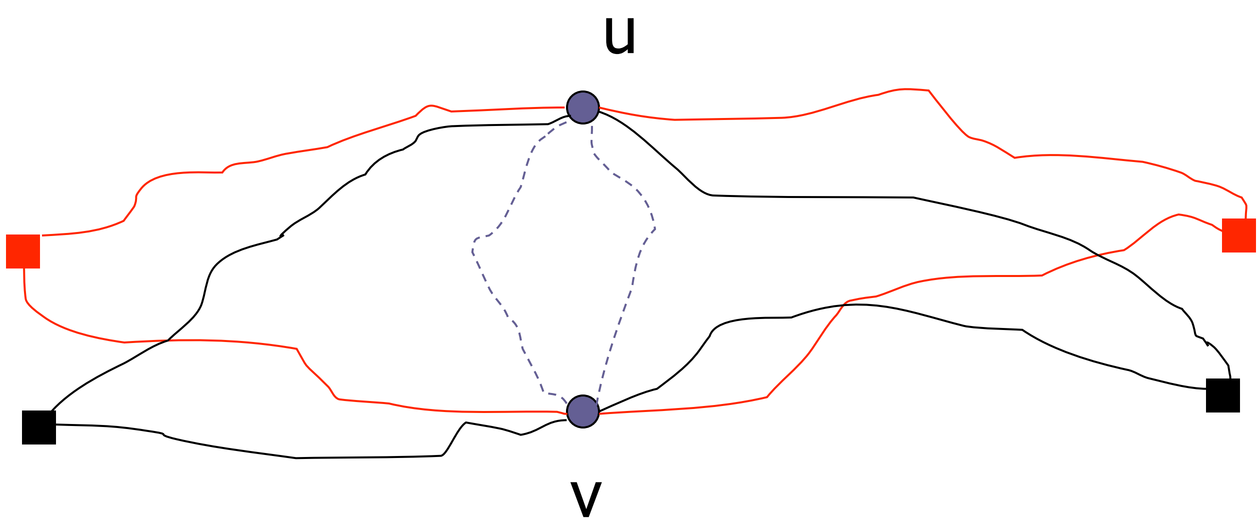
For each pair $s_i t_i$ route on k_i *disjoint paths*
(edge or node disjoint depending on applications)

Generalize SNDP (survivable network design problem)

Buy-at-Bulk with Protection

[Antonakopoulos-C-Shepherd-Zhang'07]

2-junction scheme for node-disjoint case:



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

- 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 via clustering arguments

Buy-at-Bulk with Protection

[C-Korula'08]

Single-sink with *vertex*-connectivity requirements

- $(\log n)^{O(b)}$ for b cables for $k=2$ via clustering args.
- $2^{O(\sqrt{\log h})}$ for any fixed k for non-uniform case. Algorithm is greedy inflation. Is it actually better?

[Gupta-Krishnaswamy-Ravi'10]

- $O(\log^2 n)$ for $k=2$ (*edge*-connectivity, uniform multicommodity)

Open problems

- Approximability of single-sink case for $k=2$.
  α approx. for single-sink implies $O(\alpha \text{ polylog}(n))$ for multi-comm.
- Single-sink for fixed $k>2$. Best is $2^{O(\sqrt{\log h})}$
- Multi-commodity for fixed $k>2$.

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 routing scheme [CHKS'06]
- Hardness of approximation:
  - canonical paths/girth ideas for routing problems [A'04]
- Several open problems

Uniform costs: cable model

In practice costs arise due to discrete capacity *cables*:

Cables of different type: $(c_1, u_1), (c_2, u_2), \ldots, (c_r, u_r)$

c_i: cost of cable of type i

u_i: capacity of cable of type i

$u_1 < u_2 < \ldots < u_r$ and $c_1/u_1 > c_2/u_2 > \ldots > c_r/u_r$

Can use multiple copies of each cable type

$f(x)$ = min cost set of cables of total capacity at least x