Edge-Disjoint Paths in Networks

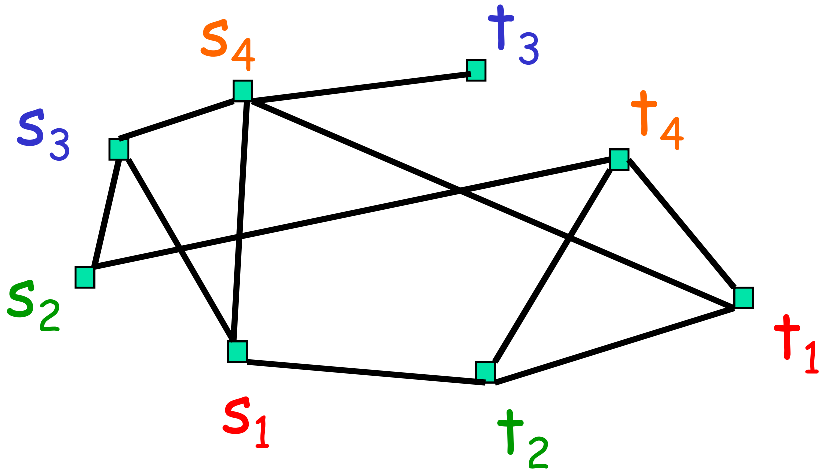
(Part 1)

Sanjeev Khanna
University of Pennsylvania

Edge Disjoint Paths Problem (EDP)

Input: Graph $G(V,E)$, source-sink pairs s_1t_1, s_2t_2,…, s_kt_k

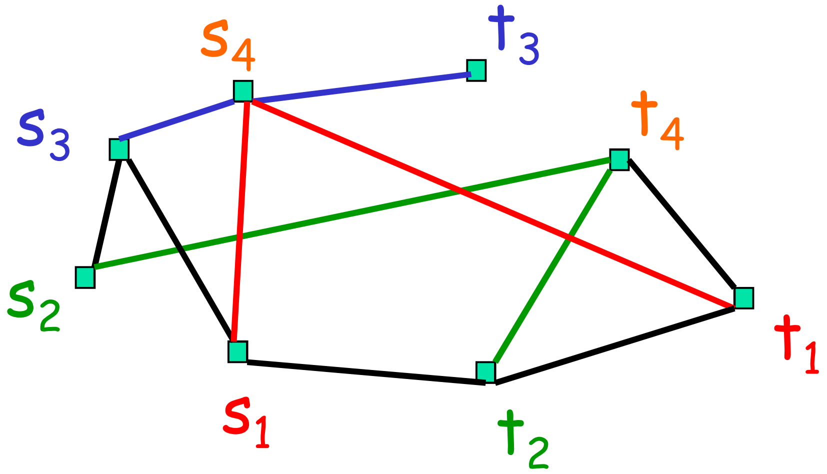
Goal: Route a maximum # of s_i-t_i pairs using edge-disjoint paths



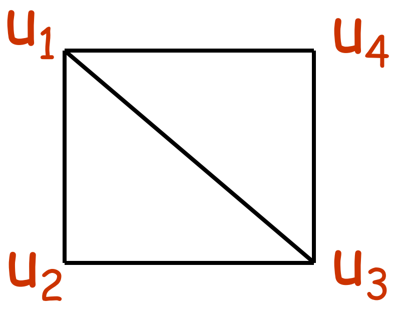
Edge Disjoint Paths Problem (EDP)

Input: Graph $G(V,E)$, source-sink pairs s_1t_1, s_2t_2,..., s_kt_k

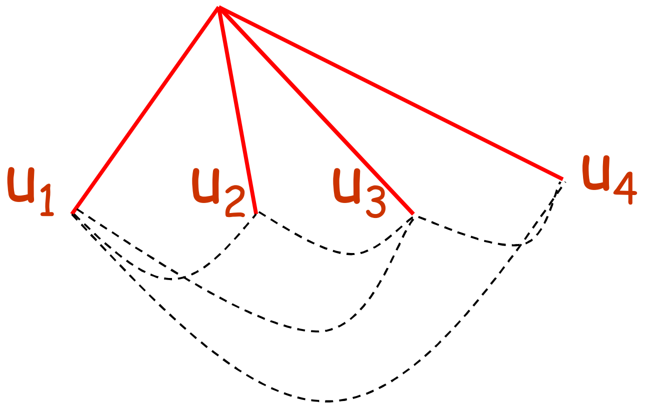
Goal: Route a maximum # of s_i-t_i pairs using edge-disjoint paths



EDP on Stars



Matching in G



EDP in a Star Graph

(Edges in G become source-sink pairs in the Star Graph)

Complexity of EDP

- EDP on star graphs is equivalent to the maximum matching problem in general graphs.
- Even for two pairs, EDP is NP-hard if G is a directed graph [Fortune, Hopcroft, Wylie'80].
- Polynomial-time solvable for constant number of pairs if G is undirected [Roberston, Seymour'88].
- NP-hard even on undirected trees when the edges have capacities.

Coping with Hardness

Settle for sub-optimal solutions: route only a fraction of the pairs that can be routed in an optimal solution.

Approximation algorithm *A*

- Runs in polynomial time
- Approximation ratio: how good is *A*
- Approx ratio α if $A(I) \geq OPT(I) / \alpha$ for all I
- Smaller the α, the closer we are to the optimal.

Overview of the Talk

- Review of classical results for EDP.
- Survey of the current state-of-the-art.
- Key algorithmic ideas underlying recent developments.
- Integrality gap and hardness results.
- Some open problems.

A Greedy Algorithm

- Among the unrouted pairs, pick the pair that has the shortest path in the current graph.
- Route this pair and remove all edges on the path from the graph.
- Repeat until no more pairs can be routed.

Clearly gives an edge-disjoint routing.

How good is this algorithm?

Analysis of the Greedy Algorithm

n: # of vertices m: # of edges

Fix an optimal solution, say, OPT.

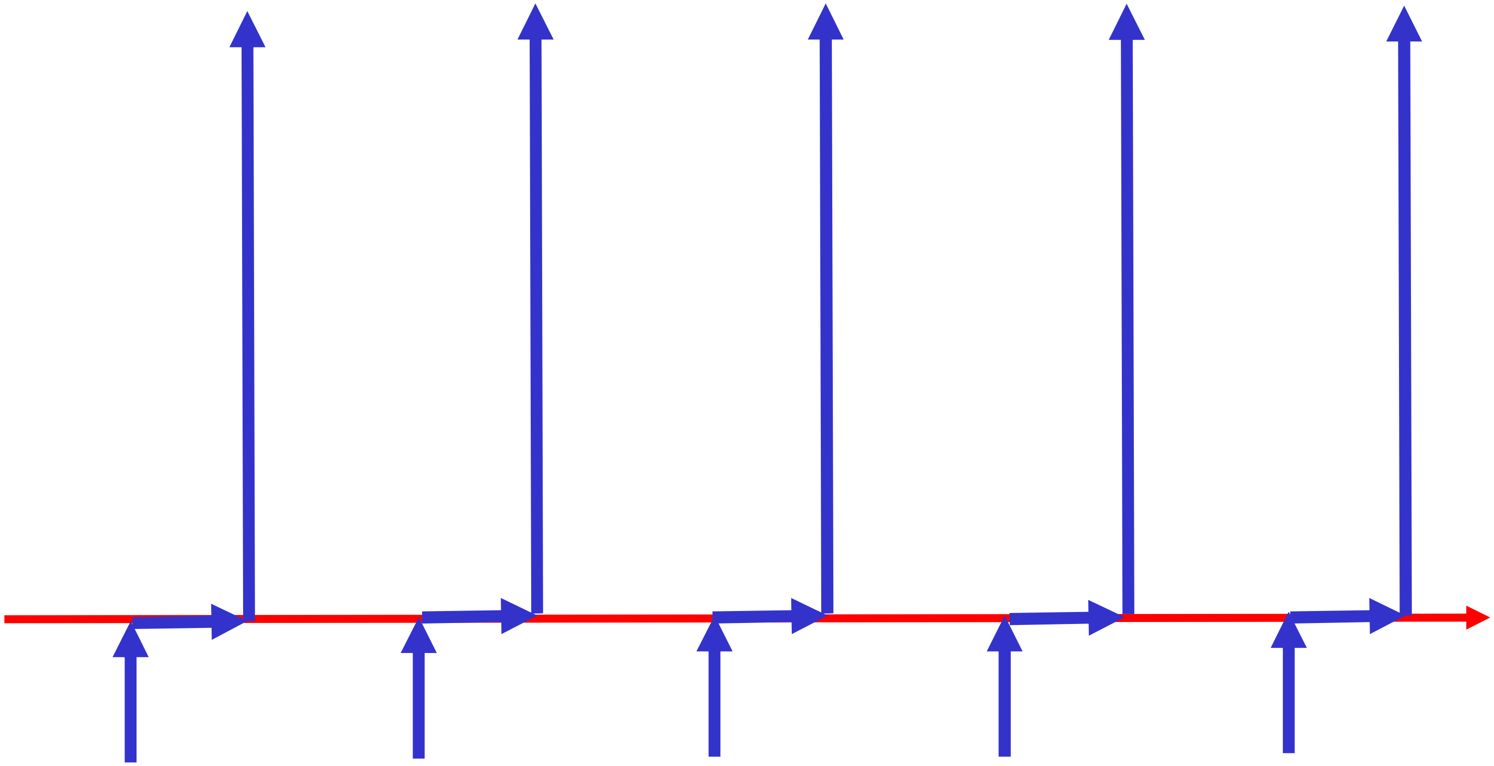
As long as the greedy path has at most $m^{1/2}$ edges, it can destroy at most $m^{1/2}$ paths in OPT.

Suppose at some point, a path chosen by greedy is longer than $m^{1/2}$. Since there are only m edges, OPT can chose at most $m^{1/2}$ paths from here on.

So greedy gives an $O(m^{1/2})$-approximation.

Can we improve it ?

Greedy chooses the red path and none of the blue pairs can be routed as a result.



Multicommodity Flow Relaxation (LP)

- Routing is relaxed to be a flow from s_i to t_i.
- A pair can be routed for a fractional amount.

x_i : amount of s_i-t_i flow that is routed.

$f(p)$: amount of flow routed on a path p.

$$\text{Max } \sum_i x_i$$

s.t.

$$\forall i \qquad x_i = \sum_{s_i\text{-}t_i \text{ paths } p} f(p)$$

$$\forall e \qquad \sum_{p:\, e \in p} f(p) \leq 1.$$

$$0 \leq x_i \leq 1.$$

A Simple Rounding Algorithm

- Among the unrouted pairs
  - Pick a pair with a shortest flow path p s.t. $f(p) > 0$.
  - Route this pair along the flow path p.
  - Discard all flow paths that share an edge with p (i.e. set $f(p') = 0$ if p' shares an edge with p).
- Repeat until no fractional flow left.

Clearly gives an edge-disjoint routing.

But how good is this algorithm?

Analysis of the Rounding Algorithm

n: # of vertices m: # of edges

Let OPT be an optimal fractional solution.

If chosen flow path p has length $\leq m^{1/2}$, routing a pair on p eliminates $\leq m^{1/2}$ units of flow from OPT.

(for every edge on p, we discard at most one unit of flow.)

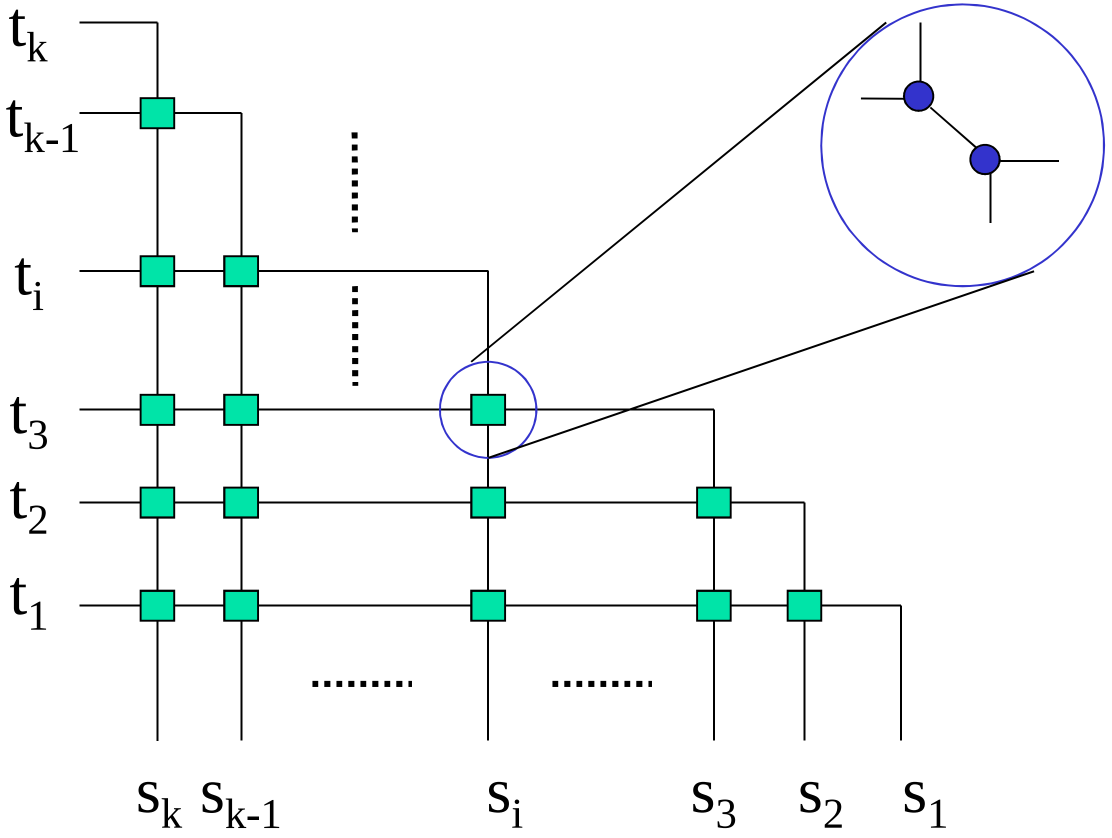
Once shortest available flow path has length $\geq m^{1/2}$, total remaining fractional flow must be $\leq m^{1/2}$.

(total capacity = m, and each unit of flow consumes $\geq m^{1/2}$ capacity.)

We get an $O(m^{1/2})$-approximation.

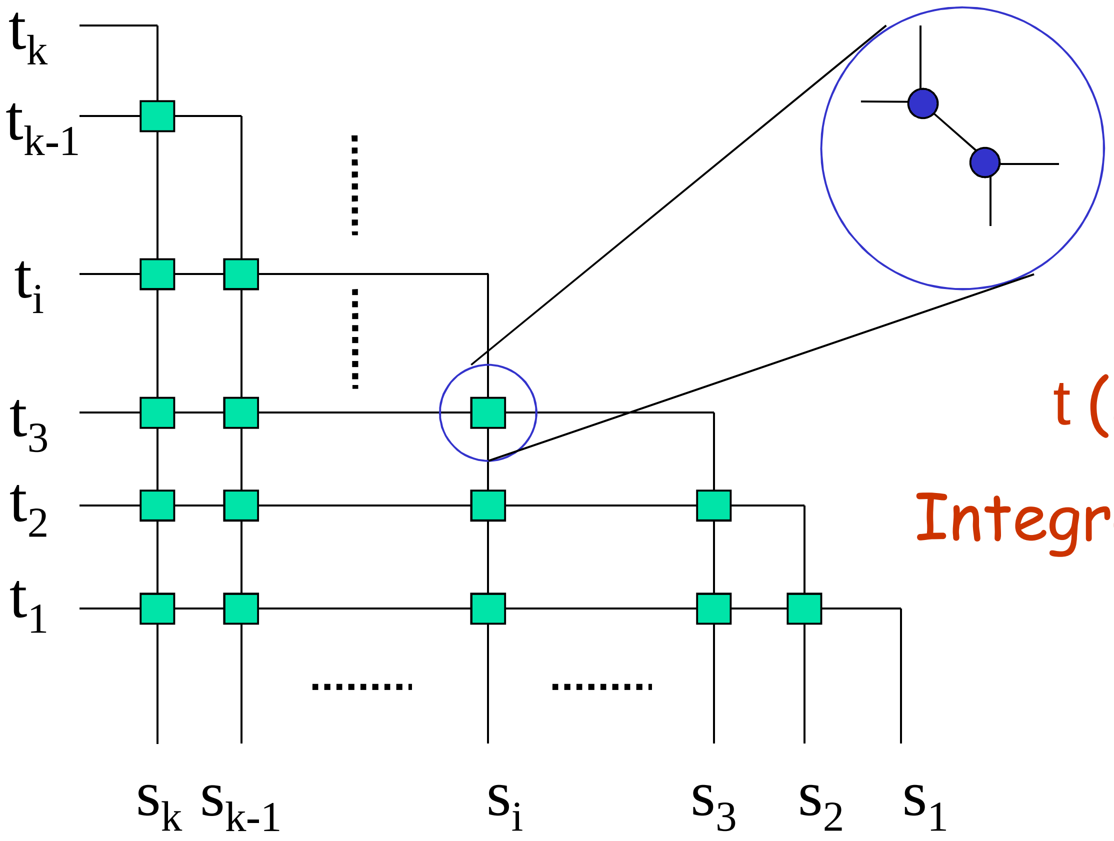
Could we do better?

[Garg, Vazirani, Yannakakis '93]



Could we do better?

[Garg, Vazirani, Yannakakis '93]



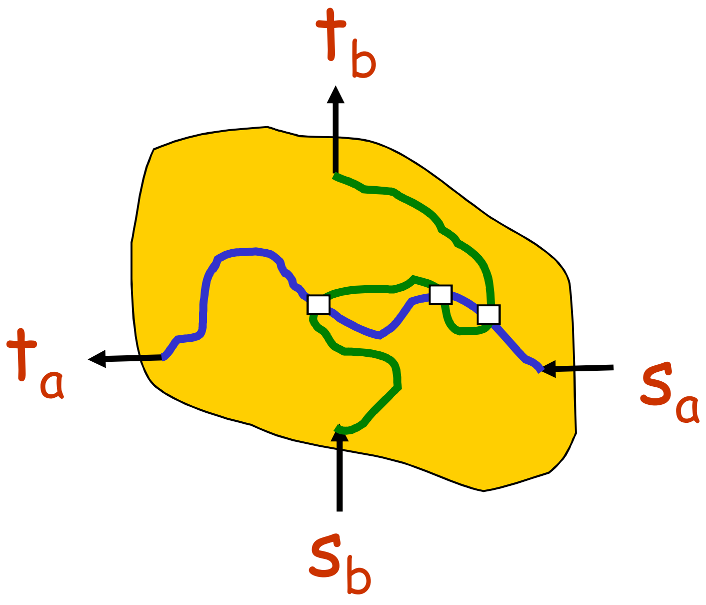
$t\,(m^{1/2})$

Integrality Gap

Gap holds for planar undirected graphs

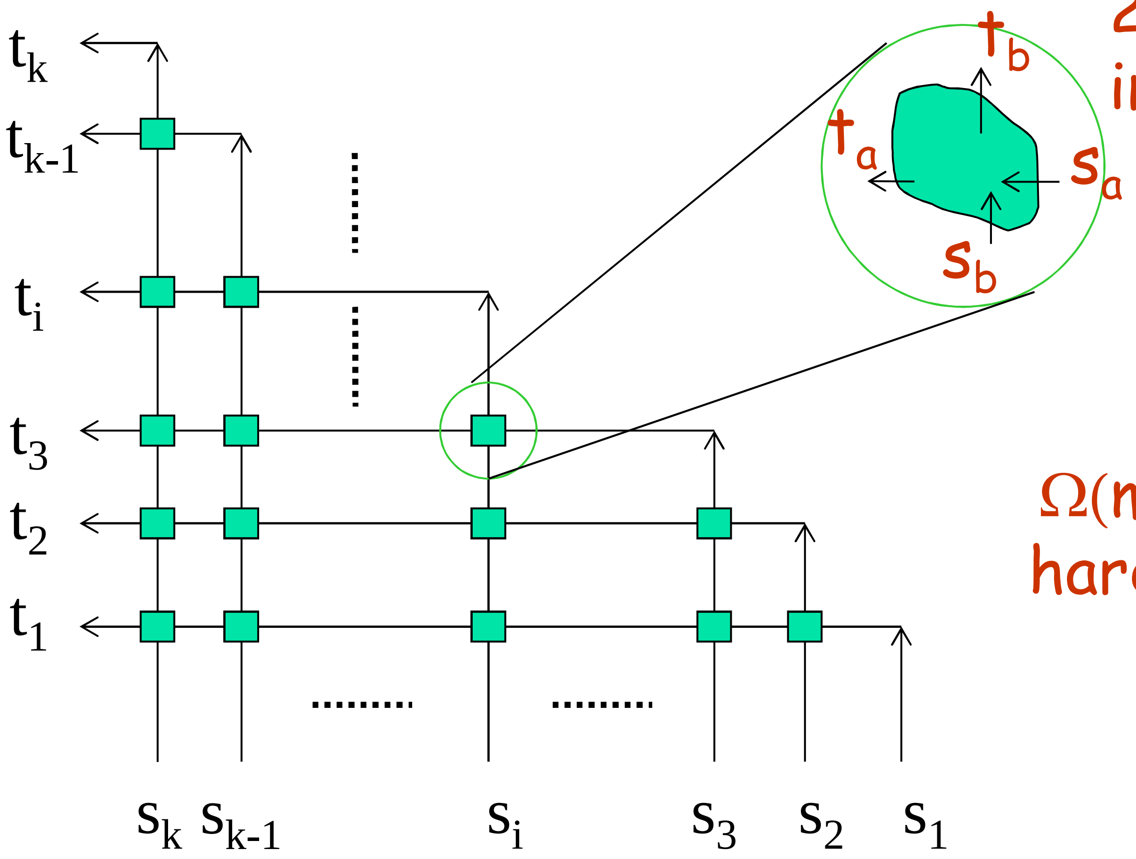
And if the Graph is Directed …

[Fortune, Hopcroft, Wylie '80] Given a directed graph G(V,E) and two pairs s_a-t_a and s_b-t_b, it is NP-hard to decide if we can route s_a to t_a and s_b to t_b on edge-disjoint paths.



Hardness of Approximation

[Guruswami, K, Rajaraman, Shepherd, Yannakakis '99]



2-pair instance G

$\Omega(m^{1/2-\varepsilon})$ hardness

EDP in Directed Acyclic Graphs

[Chekuri, K, Shepherd '06]
$O(n^{1/2})$-approximation for EDP in DAGs via rounding of the multicommodity flow relaxation.

[Chalermsook, Laekhanukit, Nanongkai '14]
$\Omega(n^{1/2-\epsilon})$ -hardness for EDP in DAGs for any $\epsilon > 0$.

What if we allow congestion …

EDP with congestion c : up to c paths can share an edge.

What happens to the integrality gap for $c \geq 2$?

Key question: does EDP become well-approximable with constant congestion?

What if we allow congestion …

Randomized Rounding [Raghavan-Thompson '87]

- Route pair (s_i,t_i) with probability x_i.
- If (s_i,t_i) is chosen, pick an s_i-t_i flow path p for routing: choose with probability proportional to $f(p)$.

$O(1)$-approximation with congestion $\Theta(\lg n/\lg\lg n)$.

[Raghavan and Thompson '87]

$O(n^{1/c})$-approximation with constant congestion c.

[Srinivasan' 97], [Baveja-Srinivasan' 00],[Azar-Regev'01].

Could we do better?

EDP with Congestion in Directed Graphs

[Andrews, Zhang '06] [Chuzhoy, Guruswami, K, Talwar '07]

- Integrality gap of the flow relaxation is roughly $n^{1/(3c)}$ for c up to $\Theta(\log n/\log\log n)$.

- Also, $n^{\Omega(1/c)}$-hardness for c up to $\Theta(\log n/\log\log n)$.

- Randomized rounding is essentially optimal for the directed edge-disjoint paths problems.

EDP with Congestion in Undirected Graphs

[Andrews, Zhang '05]

[Andrews, Chuzhoy, Guruswami, K, Talwar, Zhang '05]

- Integrality gap of the flow relaxation is at least $(\log n)^{1/(c+1)}$ with congestion c.
- $\Omega(\log^{1/(c+1)} n)$-hardness with congestion c.

Undirected Graphs: State of the Art

$\Omega(\log^{1/(c+1)} n)$-hardness with constant congestion c.

$O(n^{1/c})$-approximation with constant congestion c.

- $O(\log n)$-approximation with congestion 2 for planar graphs [Chekuri, K, Shepherd '05]
- Polylog(n) approximation with no congestion for graphs with large minimum cut [Rao-Zhou '06]
- Polylog(n) approximation with poly(lg lg n) congestion in arbitrary graphs [Andrews '10]
- Polylog(n) approximation with constant congestion in arbitrary graphs [Chuzhoy '12] [Chuzhoy-Li '12]

Well-linked Decomposition Framework for EDP

[Chekuri, K, Shepherd '04, '05]

- Start with a multicommodity flow solution but use it only to partition the graph into well-linked instances. We ignore the flow paths !
- Show that any well-linked instance contains a routing structure called a crossbar on which EDP is easy to solve.
- Route the given source-sink pairs using the crossbar.

Instance of EDP

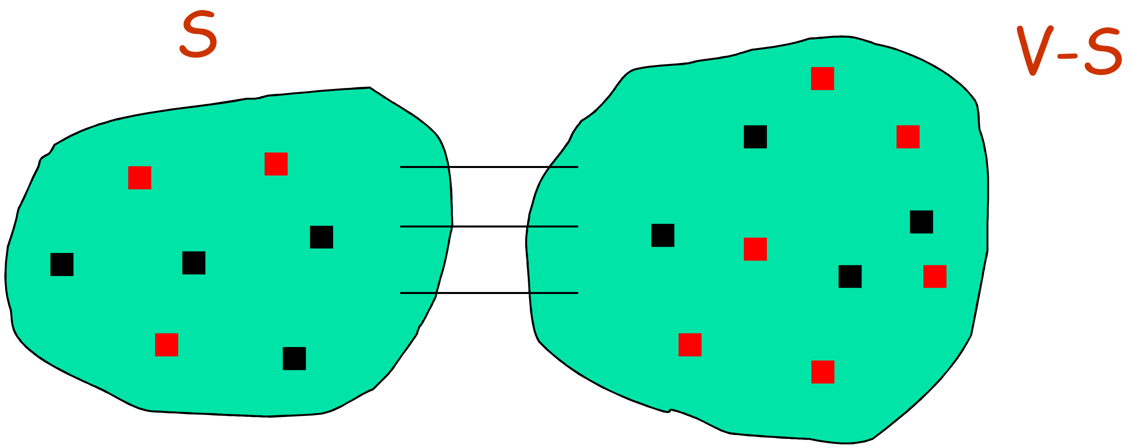
G : the underlying graph.
X : $\{s_1, t_1, s_2, t_2, \ldots, s_k, t_k\}$ is the terminal set.
Assume w.l.o.g. that s_i, t_i are distinct and have degree 1 in the graph.

The goal is to route a given matching on X.

Also assume w.l.o.g. that degree of any vertex in G is bounded by 4.

Well-linked Set

Subset X is well-linked in G if for any partition (S,V-S):
# of edges cut ≥ # of X vertices in the smaller side.



$\forall\, S \subset V$ s.t. $|S \cap X| \leq |X|/2$, $|E(S, V\text{-}S)| \geq |S \cap X|$.

Instance of EDP

G : the underlying graph.

X : $\{s_1, t_1, s_2, t_2, \ldots, s_k, t_k\}$ is the terminal set.

Assume w.l.o.g. that s_i, t_i are distinct.

The goal is to route a given matching on X.

Well-linked Instance of EDP

G : the underlying graph.

X : $\{s_1, t_1, s_2, t_2, \ldots, s_k, t_k\}$ is the terminal set.

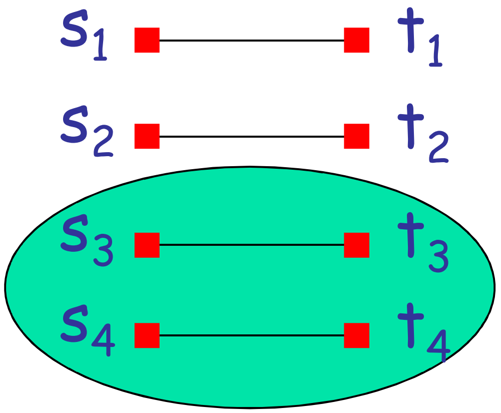
Assume w.l.o.g. that s_i, t_i are distinct.

The goal is to route a given matching on X.

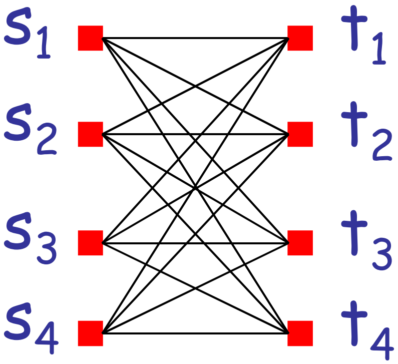
X is well-linked in G.

Theorem [Chekuri, K, Shepherd '05] Any instance of EDP can be reduced to a collection of well-linked instances with only a polylog(n) factor loss in the solution value.

Examples

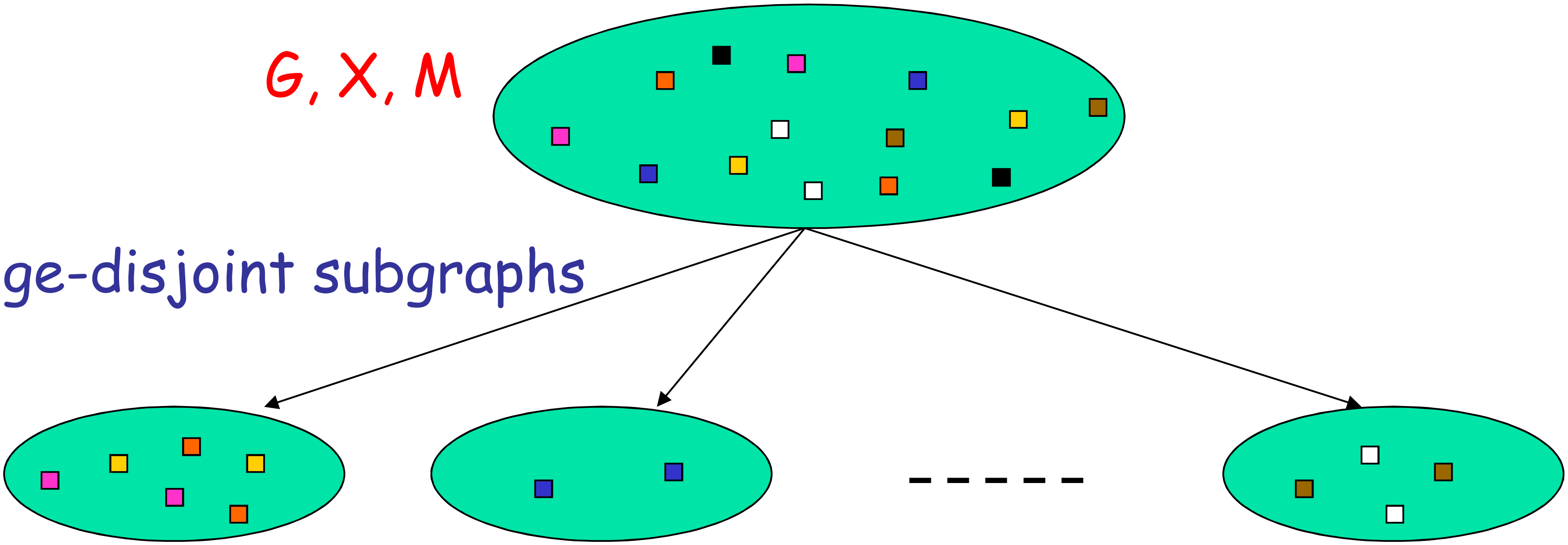


Not a well-linked instance



A well-linked instance

Well-linked Decomposition



G, X, M

Edge-disjoint subgraphs

G_1, X_1, M_1 G_2, X_2, M_2 G_r, X_r, M_r

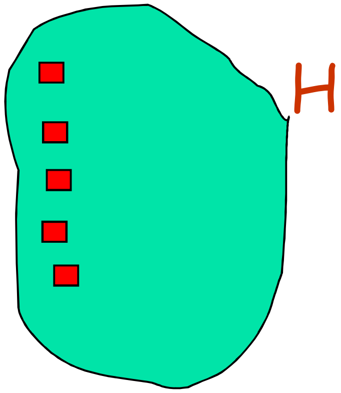
$M_i \subset M$

X_i is well-linked in G_i

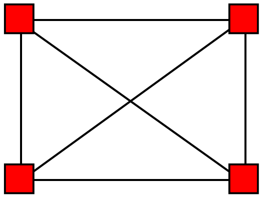
$\sum_i |X_i| \geq$ OPT/polylog(k)

Crossbars

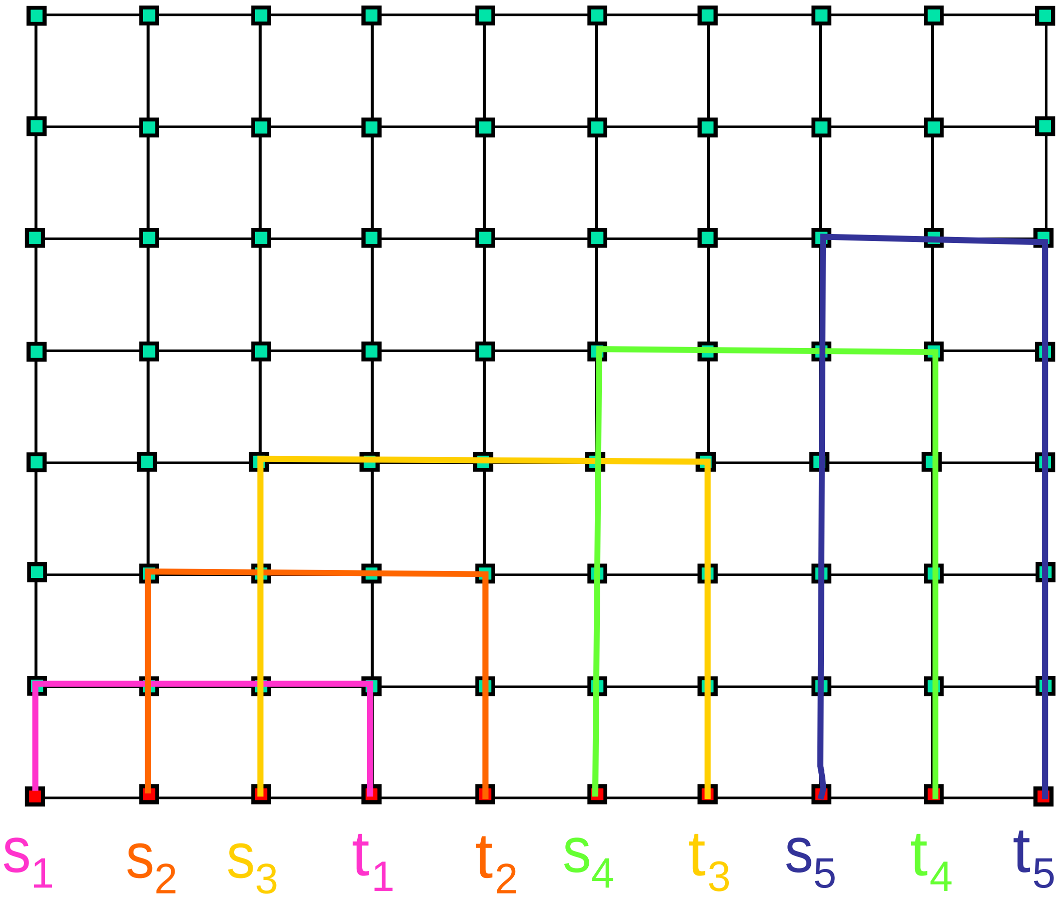
$H(V,E)$ is a cross-bar with respect to an interface $I \subseteq V$ if any matching on I can be routed using edge-disjoint paths.



Ex: a complete graph is a cross-bar with $I=V$



Grids as Crossbars



First row is the interface

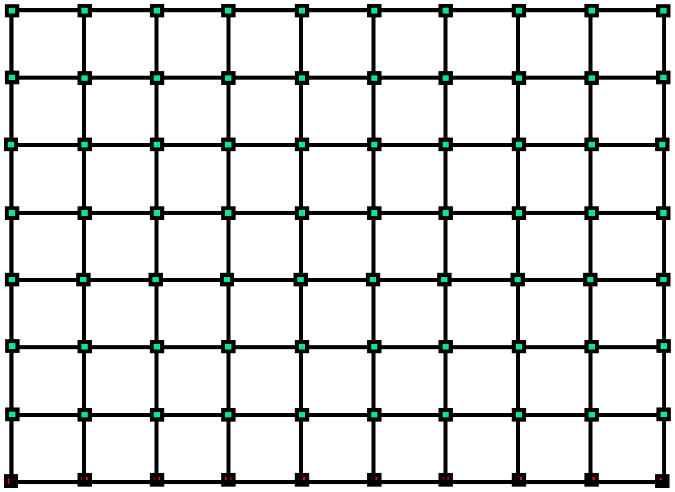
Application: EDP in Planar Graphs

- Solve the multicommodity flow relaxation.
- Use the solution to partition the given instance into planar well-linked instances.
- Find a crossbar in each well-linked instance, and route using the crossbar.

Planar Well-linked Instances

Theorem [Robertson, Seymour, Thomas '94]
If G is a planar graph with k well-linked terminals, then with congestion 2, an $\Omega(k) \times \Omega(k)$ grid H can be embedded in G.

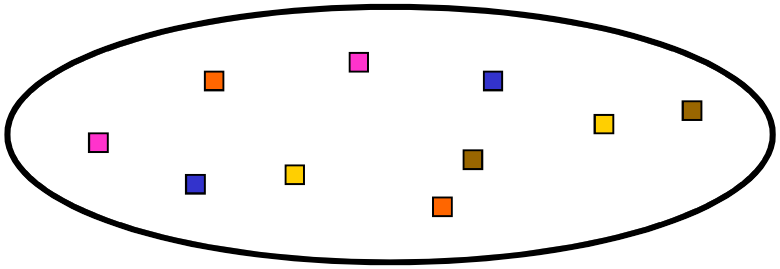
Routing pairs in X using H



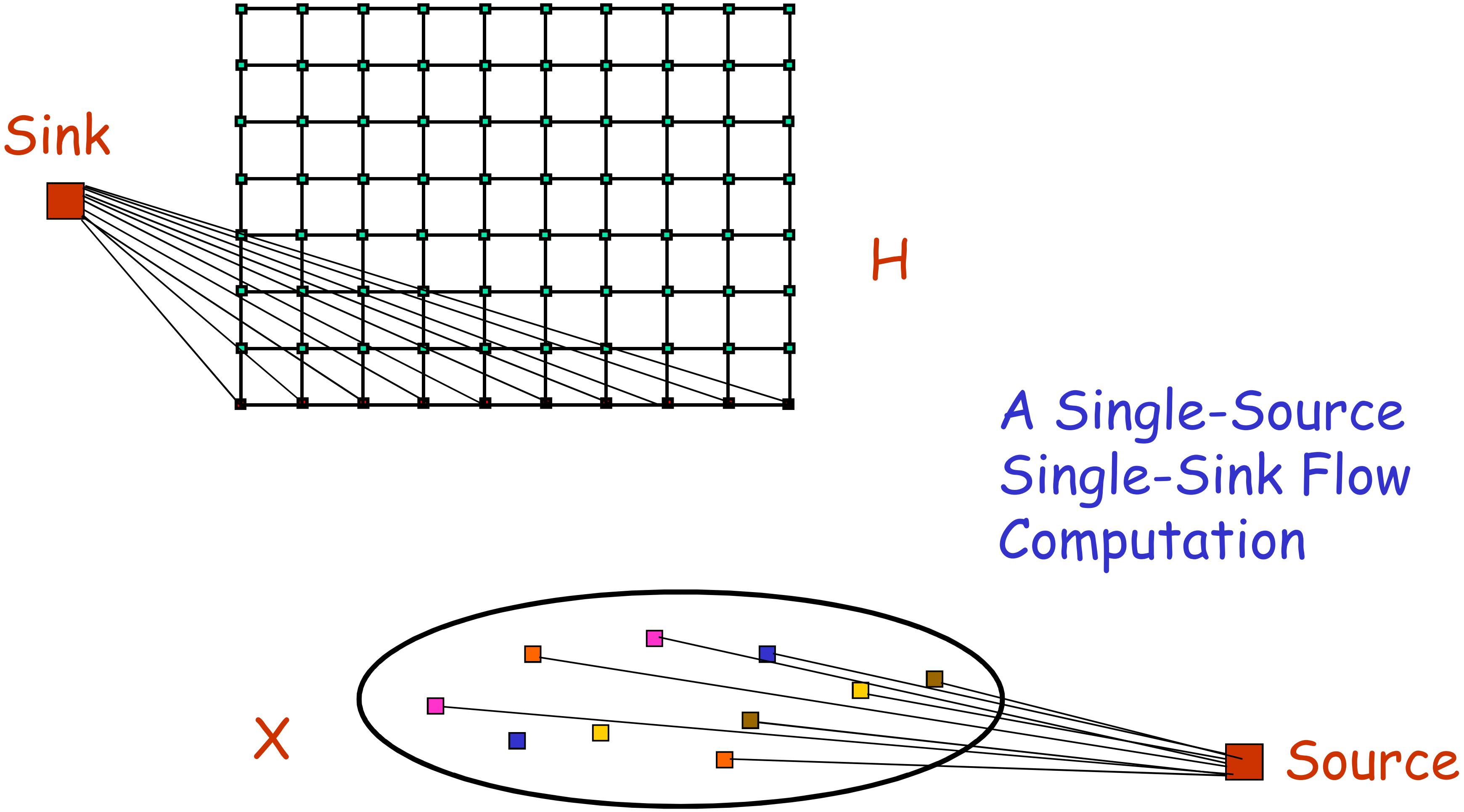
Grid H

Interface I

Terminals X

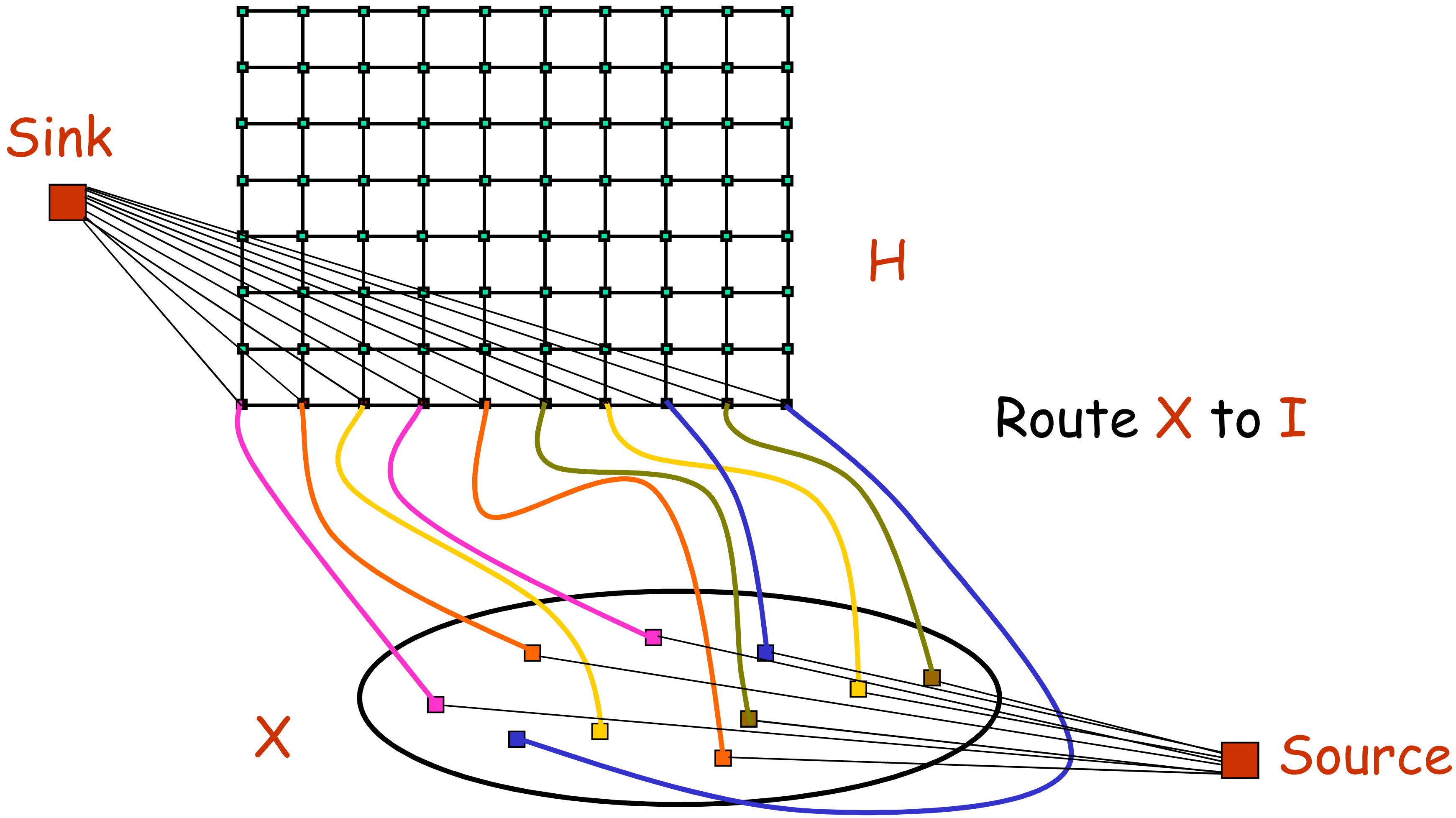


Routing pairs in X using H

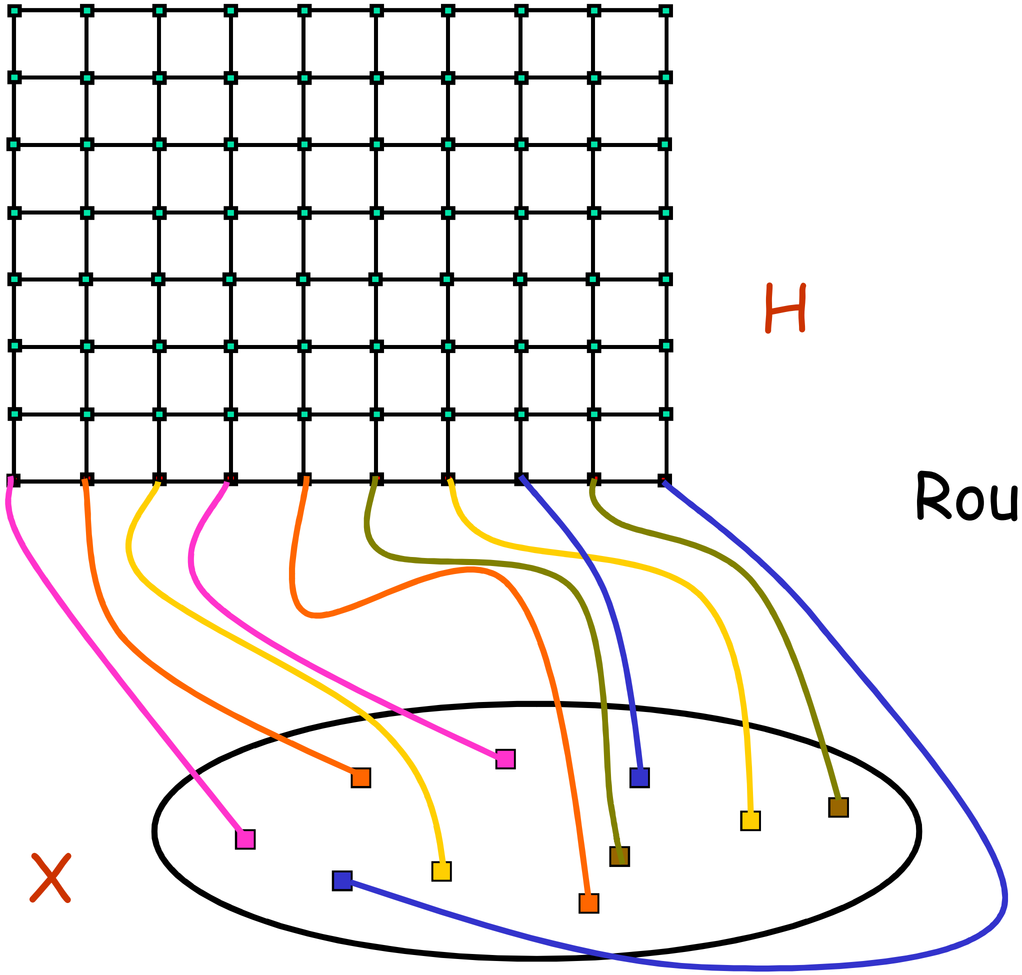


A Single-Source Single-Sink Flow Computation

Routing pairs in X using H

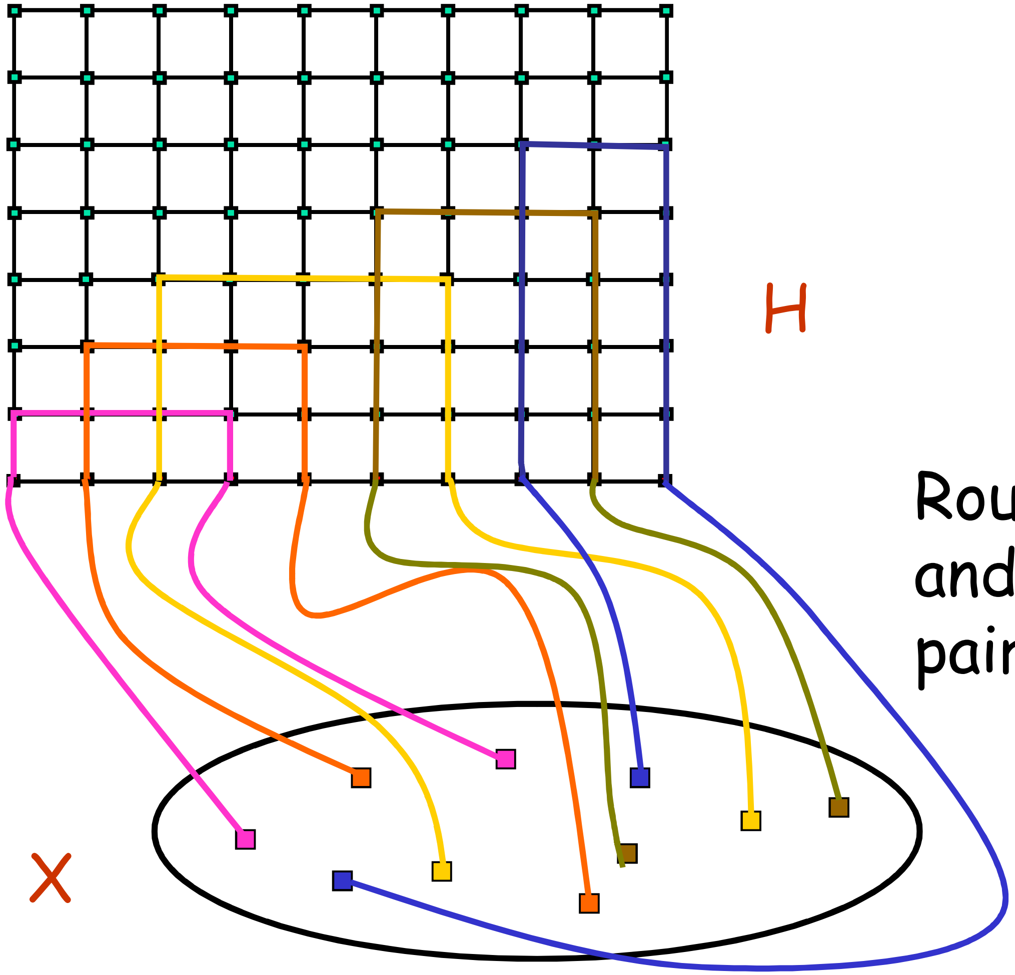


Routing pairs in X using H



Route X to I

Routing pairs in X using H



Route X to I and use H for pairing up

EDP in Planar Graphs

Theorem [Chekuri, K, Shepherd '05]

EDP in planar graphs can be approximated to within a factor of $O(\log n)$ with congestion 2.

- Prior to this, only $n^{\Omega(1)}$-approximation was known for any constant congestion.
- Recall that the integrality gap is $\Omega(n^{1/2})$ when no congestion is allowed.
- Later, $O(1)$-approximation with congestion 4, and then $O(1)$-approximation with congestion 2 [Chekuri, K, Shepherd '06; Seguin-Charbonneau, Shepherd '11].

Well-linked Decomposition

Flow Well-Linked Sets

A subset X is flow-well-linked in G if the following multicommodity flow is feasible in G:

for u,v in X, $d(uv) = 1/|X|$

An instance of product multicommodity flow on X.

Cut vs Flow Well-Linked Sets

X flow-linked $\Rightarrow$ X is cut-linked

X cut-linked $\Rightarrow$ X is flow-linked with congestion $\beta(G)$

$\beta(G)$ - flow-cut gap for product multicommodity instances in G

Fractional Version

π: a non-negative weight function on X

$\pi(v)$: weight of v in X

X is π-cut-linked: for all $S \subseteq V$ with $\pi(S \cap X) \leq \pi(X)/2$,
$|E(S,V-S)| \geq \pi(S \cap X)$

X is π-flow-linked: multicommodity flow instance with
$d(uv) = \pi(u)\ \pi(v)\ /\ \pi(X)$ is feasible in G

Well-linked Instance

G : the underlying graph.

X : $\{s_1, t_1, s_2, t_2, \ldots, s_k, t_k\}$ is the terminal set.

Assume w.l.o.g. that s_i, t_i are distinct.

The goal is to route a given matching on X.

X is well-linked in G.

Fractional Well-linked Instance

G : the underlying graph.
X : $\{s_1, t_1, s_2, t_2, \ldots, s_k, t_k\}$ is the terminal set.
Assume w.l.o.g. that s_i, t_i are distinct.
The goal is to route a given matching on X.

X is π-well-linked in G, and for each pair $s_{j,} t_j$ we have $\pi(s_j) = \pi(t_j)$.

Assume that for $0 \leq \pi(v) \leq 1$ all $v \in X$.

Decomposition using Sparse Cuts

We now describe the process for creating fractional flow well-linked instances.

Start with the LP solution for the given instance.

f_j : flow for pair $s_j t_j$.
$f = \sum_i f_j$ is the total flow in LP.

Define π to be $\pi(s_j) = \pi(t_j) = f_j$.

Decomposition Algorithm

$\beta(G)$ – flow-cut gap in G

```
If $X$ is $\pi/(10\ \beta(G) \log k)$-flow-linked then stop;

Else
    Find an approximate sparse cut $(S,V-S)$ w.r.t. $\pi$ in $G$
    Remove flow on edges of the cut $(S,V-S)$
    $G_1 = G[S]$, $G_2 = G[V-S]$
    Recurse on $G_1$, $G_2$ with the remaining flow
```

Analysis

Suppose the remaining graphs at end of recursion are:
(G_1,X_1,π_1) , (G_2,X_2,π_2) , ..., (G_r,X_r,π_r)

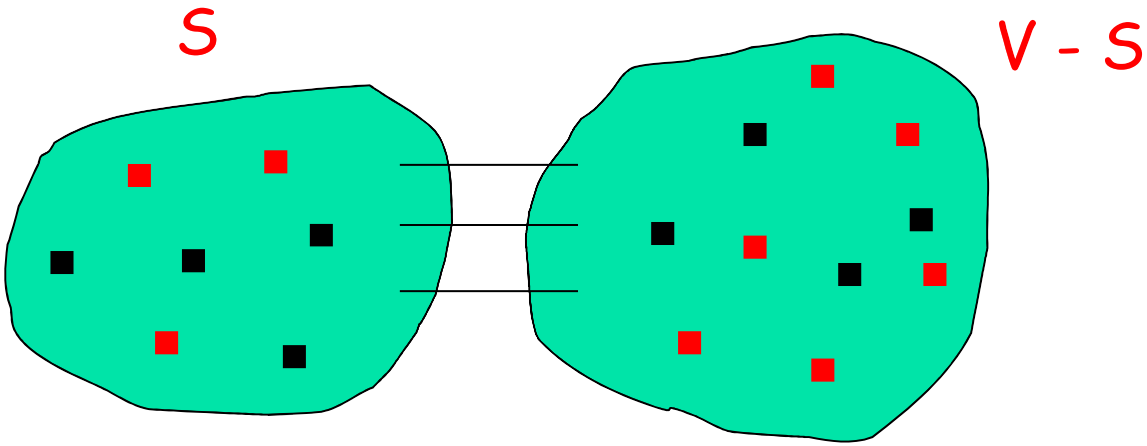
π_i is the remaining flow for X_i
X_i is $\pi_i/(10\ \beta(G) \log k)$ flow-linked in G_i

ġ$_i$ $\pi_i(X_i) \geq$ (Original flow) – (# of edges cut)

Bounding the # of Edges Cut

X is not $\pi/(10\ \beta(G) \log k)$ flow-linked

$\Rightarrow$ # of edges in the cut $(S, V-S) \leq \pi(S)/(10 \log k)$



Analysis Continued …

Claim: total number of edge cut is at most $f/2$.

$T(x)$: max # of edges cut if started with flow x

$T(f) \leq T(f_1) + T(f_2) + f_1 / (10 \log k)$
$\Rightarrow T(f) \leq f/2.$

Thus $ġ_i \ \pi_i(X_i) \geq f/2.$

Each X_i is $\pi_i/(10\ \beta(G) \log k)$ flow-well-linked.

Fractional to Integer Well-linked

Theorem[Chekuri, K, Shepherd '05]

Given an input instance G, X, M where X is π-flow well-linked, we can recover G, X', M' such that

- X' is $\frac{1}{2}$-flow well-linked,
- $|X'| = \Omega(\pi(X))$, and
- $M' \subseteq M$, is a matching defined over X'.

Proof Idea: Use a spanning tree to cluster fractional mass into integral units.

A similar result can be shown for cut well-linked.

Thank You!