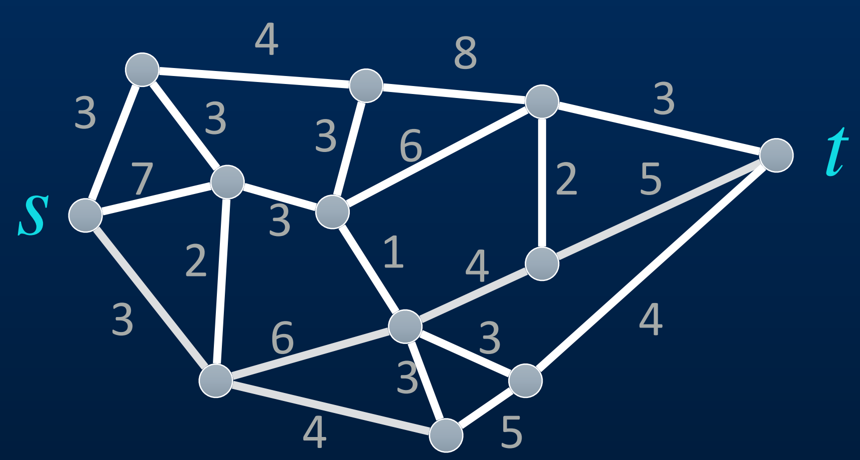
APMF < APSP?

Gomory-Hu Tree in Subcubic Time



Which are easier to compute: distances or connectivities?

Shortest-Path$(s, t) = ?$

Max-Flow$(s, t) = ?$

With **Robert Krauthgamer** (Weizmann) and **Ohad Trabelsi** (Michigan)

[SODA'20, FOCS'20, STOC'21, FOCS'21, SODA'22]

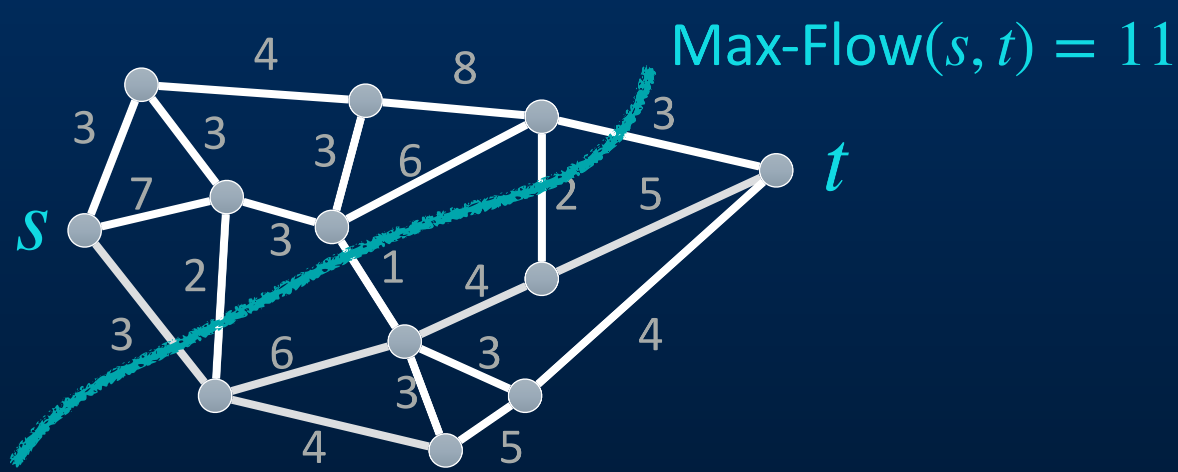
+ *[new paper]* also with **Jason Li** (Simons), **Debmalya Panigrahi** (Duke), and **Thatchaphol Saranurak** (Michigan)

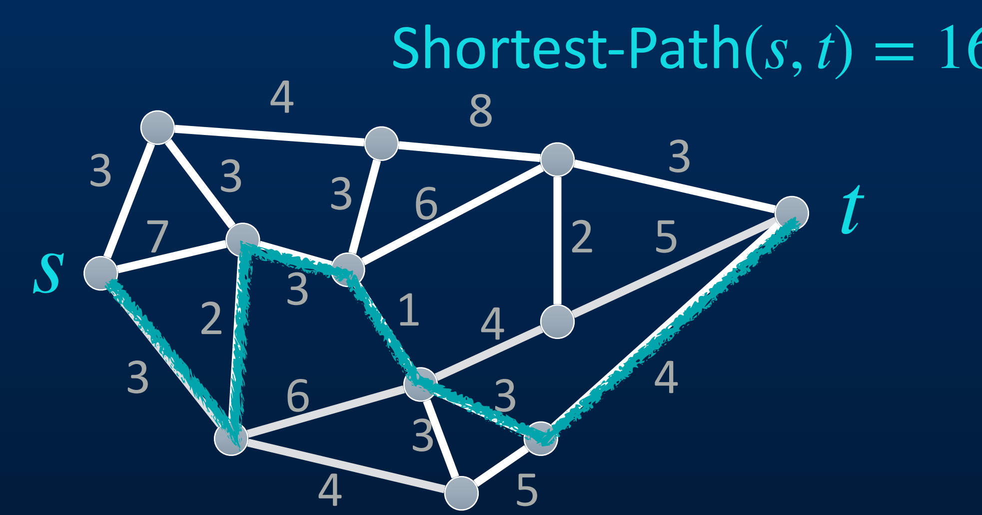
Amir Abboud (Weizmann)

Max-Flow vs. Shortest Path

= Min-s,t-Cut

Which is easier to compute?





[Dinitz'70] mn

[Goldberg-Rao'98] $mn^{2/3}$

[Lee-Sidford'14] $mn^{1/2}$

via Continuous Optimization

[BLLSSSW'21] $\tilde{O}(m + n^{1.5})$

1950's Dijkstra's

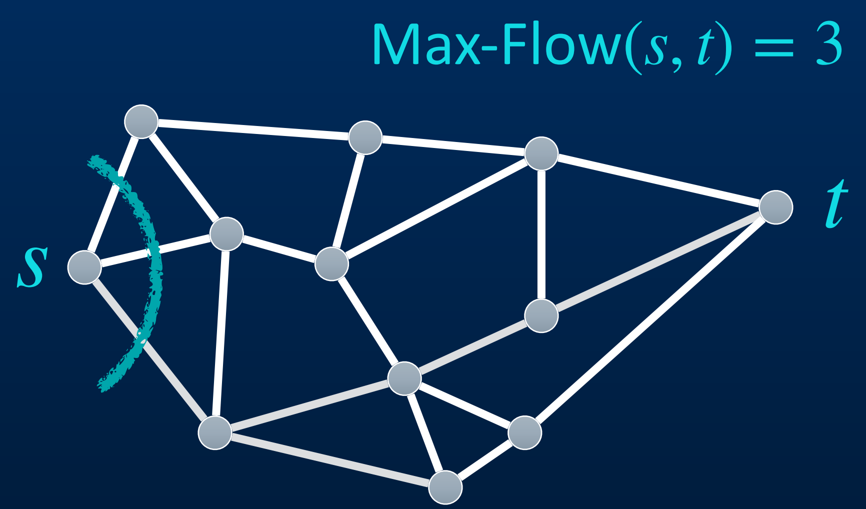
$O(m + n \log n)$

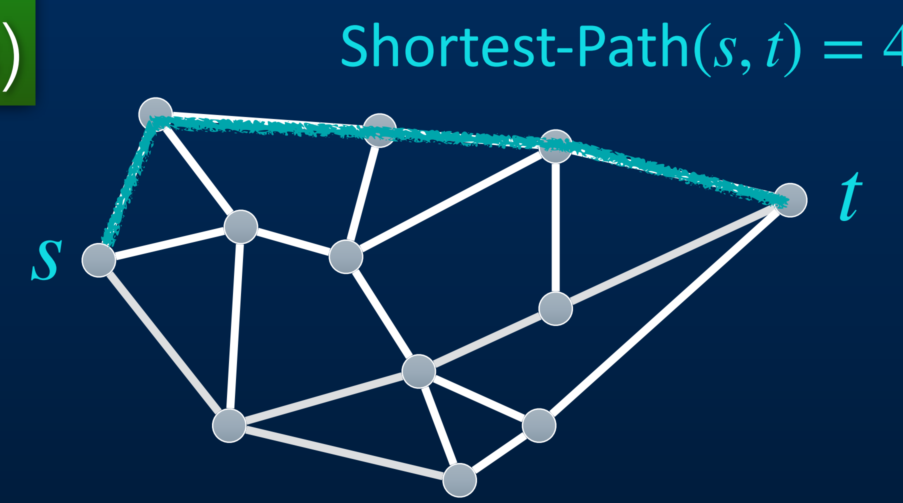
Both $\tilde{\Theta}(n^2)$ but shortest path is much simpler.

Max-Flow vs. Shortest Path

Which is easier to compute?

Unweighted (simple graphs)





via Randomized Contractions

[Karger-Levine'02] $\tilde{O}(m + n^2)$

via Continuous Optimization

[BLLSSSW'21] $\tilde{O}(m + n^{1.5})$

1950's BFS

$O(m + n)$

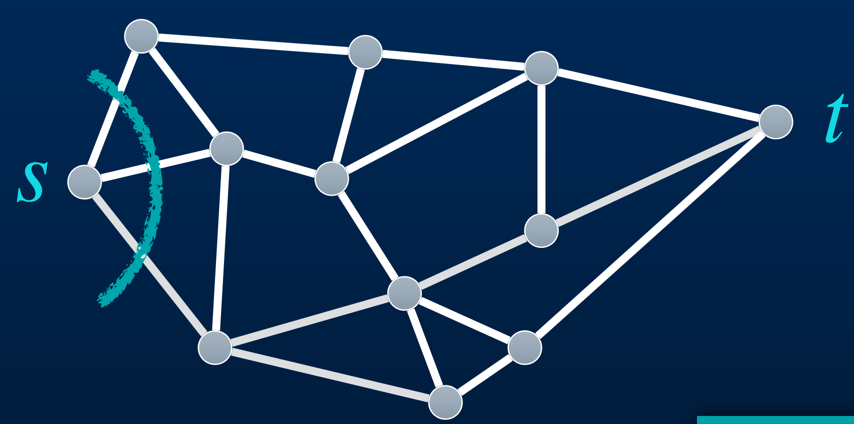
Both $\tilde{\Theta}(n^2)$ *but shortest path is much simpler.*

Max-Flow vs. Shortest Path

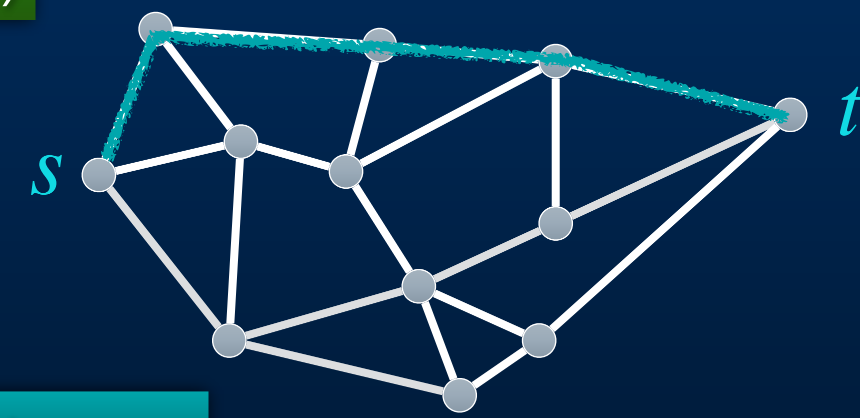
Which is easier to compute?

Unweighted (simple graphs)

Max-Flow$(s, t) = 3$



Shortest-Path$(s, t) = 4$



Both work for directed graphs too.

via Randomized Contractions

[Karger-Levine'02] $\tilde{O}(m + n^2)$

via Continuous Optimization

[BLLSSSW'21] $\tilde{O}(m + n^{1.5})$

1950's BFS

$O(m + n)$

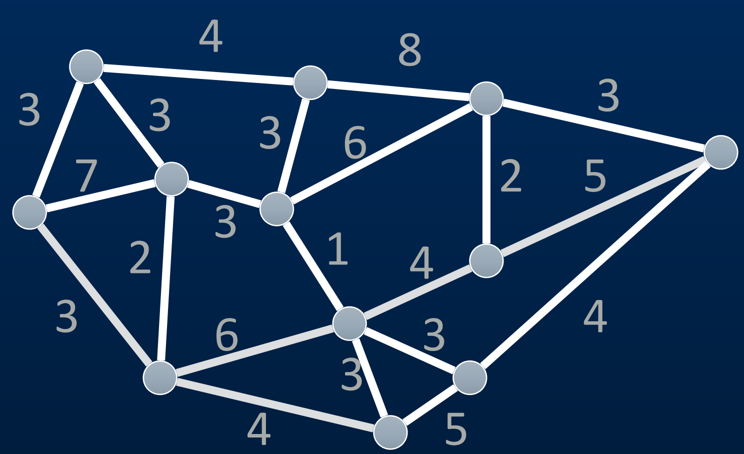
Also solves the Single-Source version.

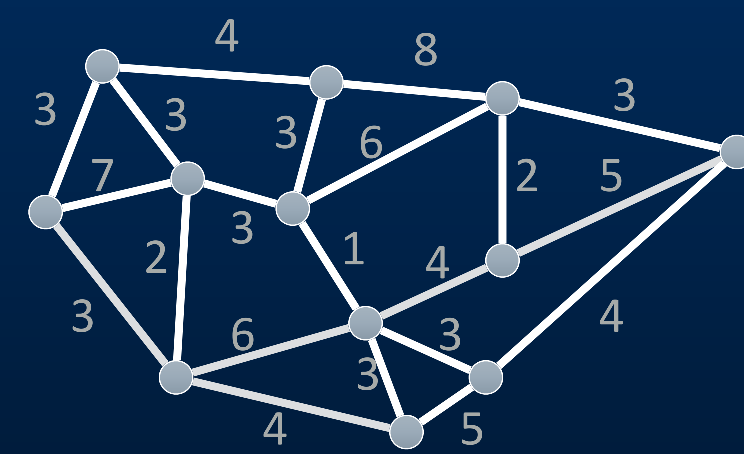
Both $\tilde{\Theta}(n^2)$ *but shortest path is much simpler.*

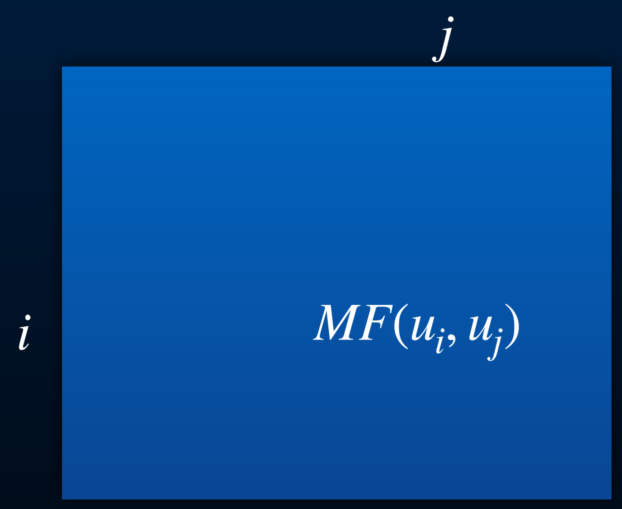
All-Pairs MF vs. All-Pairs SP

APMF: $\forall s, t \in V$: Max-Flow$(s, t) = ?$

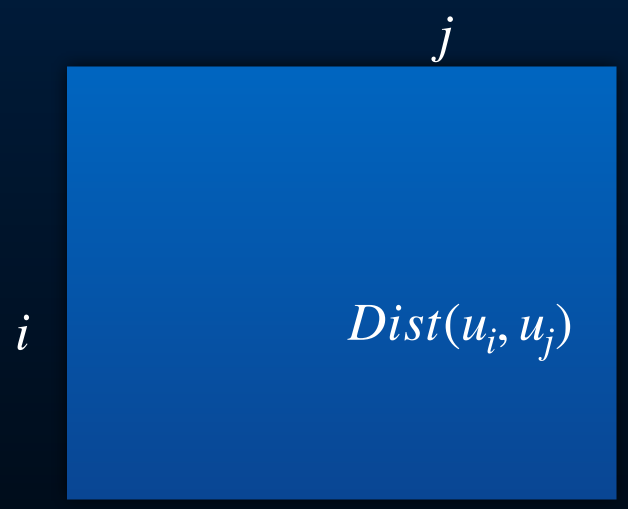
APSP: $\forall s, t \in V$: Shortest-Path$(s, t) = ?$







Just the values.
Output size $\tilde{O}(n^2)$

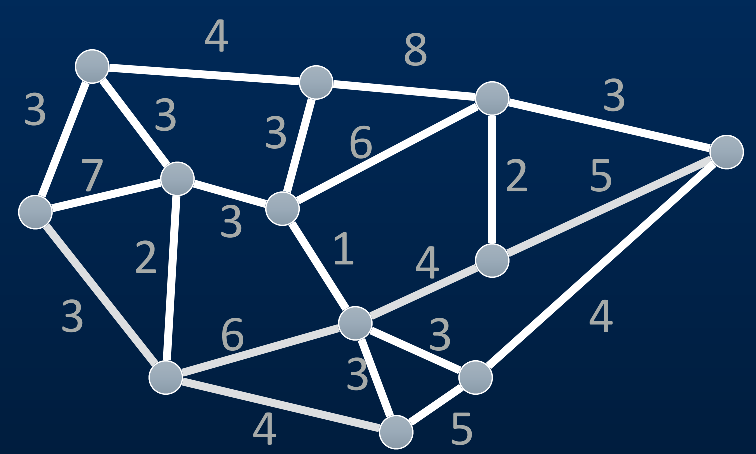


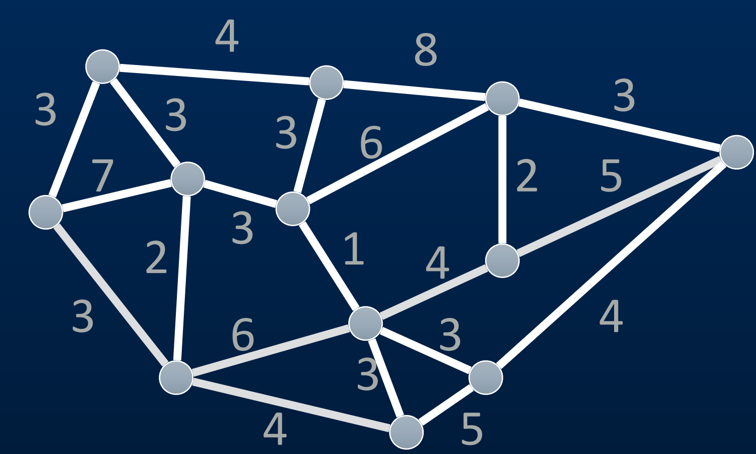
Which is easier to compute?

All-Pairs MF vs. All-Pairs SP

APMF: $\forall s,t \in V$: Max-Flow$(s,t) = ?$

APSP: $\forall s,t \in V$: Shortest-Path$(s,t) = ?$





Trivial: $n^2 \cdot MF(n) = \tilde{O}(n^4)$ $\quad$ $n^2 \cdot SP(n) = \tilde{O}(n^4)$

Trivial 2: $\quad$ $n \cdot$ Single-Source-SP$(n) = \tilde{O}(n^3)$

Gomory-Hu 1961:

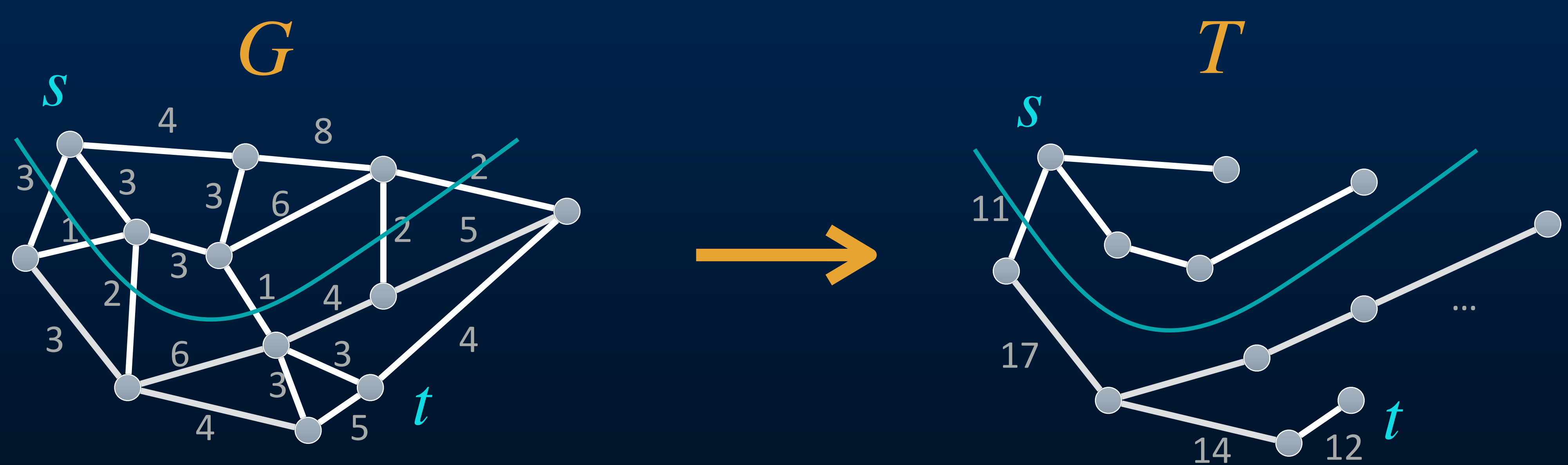
$$(n-1) \cdot MF(n) = \tilde{O}(n^3)$$

Which is easier to compute?

Gomory-Hu Tree

Thm [GH 1961]:

Every undirected graph has a (weighted) *cut-equivalent tree*.
Moreover, it can be computed in $(n-1)\cdot MF(n)$ time.



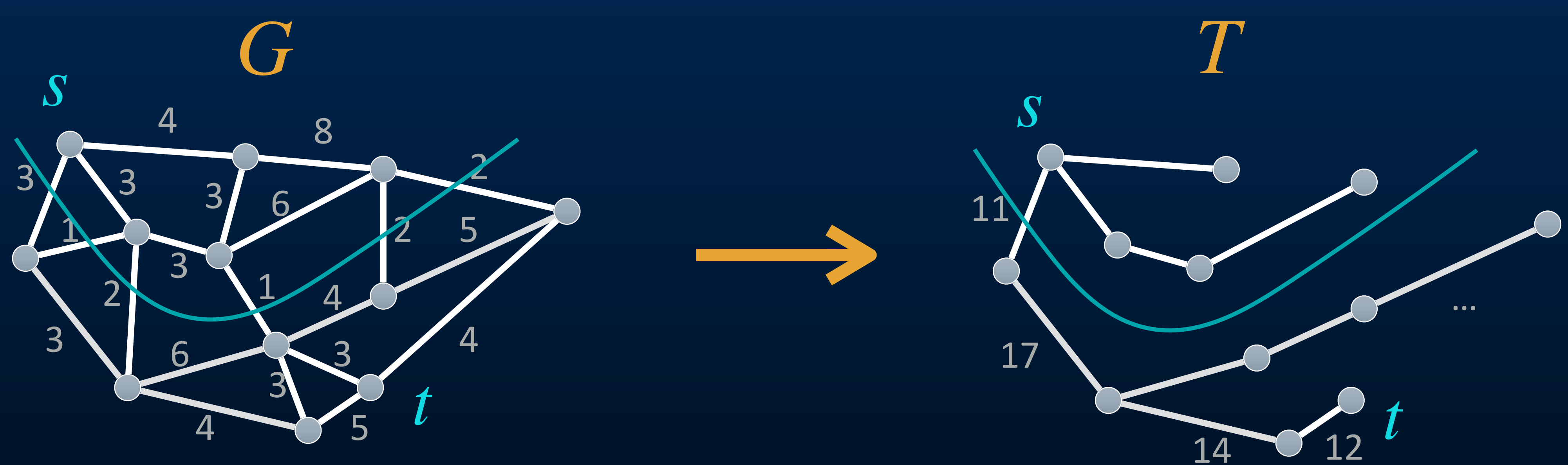
$$\forall s,t \in V: \text{Min-Cut}_G(s,t) = \text{Min-Cut}_T(s,t)$$

APMF on a tree is in $\tilde{O}(n^2)$ time. *The challenge is to compute a GH tree…*

Gomory-Hu Tree

Thm [GH 1961]:

Every undirected graph has a (weighted) *cut-equivalent tree*.
Moreover, it can be computed in $(n-1) \cdot MF(n)$ time.



$$\forall s,t \in V: \text{Min-Cut}_G(s,t) = \text{Min-Cut}_T(s,t)$$

- => APMF has only n-1 answers.
- Space-optimal min-cut oracle.
- First graph sparsification result?

Impossible for APSP.

All-Pairs MF vs. All-Pairs SP

Gomory-Hu 1961:

$(n-1) \cdot MF(n) = \tilde{O}(n^3)$

Open: $o(n) \cdot MF(n)$?

Meanwhile…

APSP in "mildly sub-cubic time"

Author	Runtime	Year
Fredman	$n^3 \log \log^{1/3} n / \log^{1/3} n$	1976
Takaoka	$n^3 \log \log^{1/2} n / \log^{1/2} n$	1992
Dobosiewicz	$n^3 / \log^{1/2} n$	1992
Han	$n^3 \log \log^{5/7} n / \log^{5/7} n$	2004
Takaoka	$n^3 \log \log^2 n / \log n$	2004
Zwick	$n^3 \log \log^{1/2} n / \log n$	2004
Chan	$n^3 / \log n$	2005
Han	$n^3 \log \log^{5/4} n / \log^{5/4} n$	2006
Chan	$n^3 \log \log^3 n / \log^2 n$	2007
Han, Takaoka	$n^3 \log \log n / \log^2 n$	2012
Williams	$n^3 / 2^{\Omega(\sqrt{\log n})}$	2014

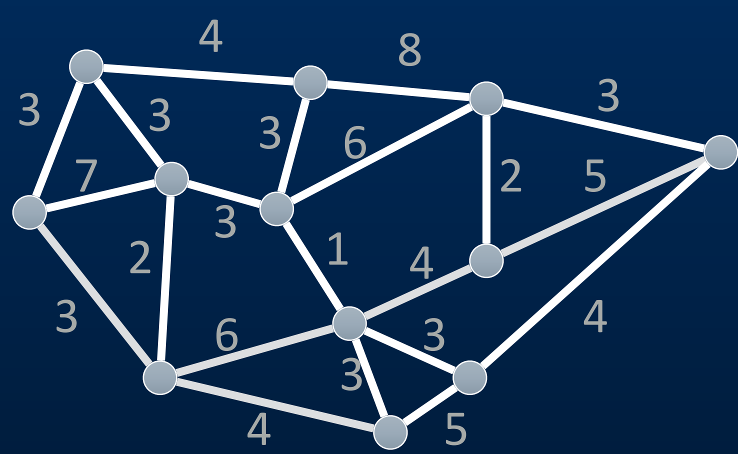
Both $\tilde{O}(n^3)$ but APSP seems easier.

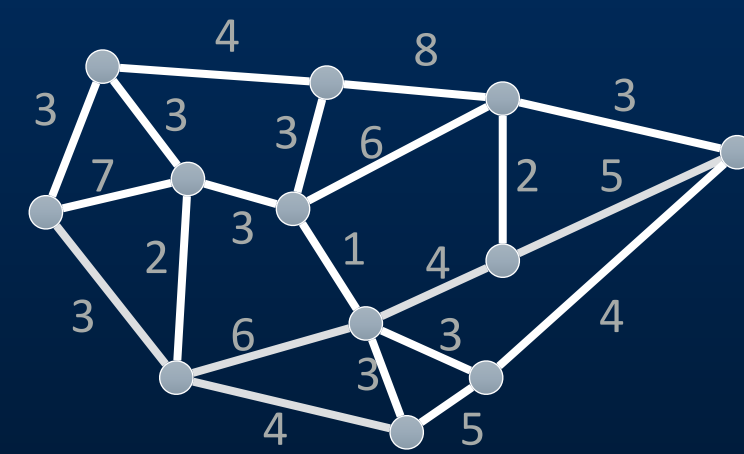
All-Pairs MF vs. All-Pairs SP

Which is easier to compute?

APMF: $\forall s,t \in V : \text{Max-Flow}(s,t) = ?$

APSP: $\forall s,t \in V : \text{Shortest-Path}(s,t) = ?$





Trivial: $n^2 \cdot MF(n) = \tilde{O}(n^4)$

$n^2 \cdot SP(n) = \tilde{O}(n^4)$

Trivial 2:

$n \cdot SSSP(n) = \tilde{O}(n^3)$

Directed too.

Gomory-Hu 1961:

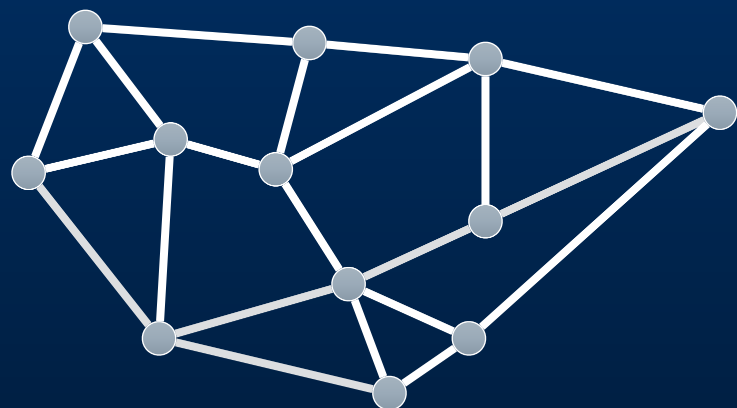
Undirected only.

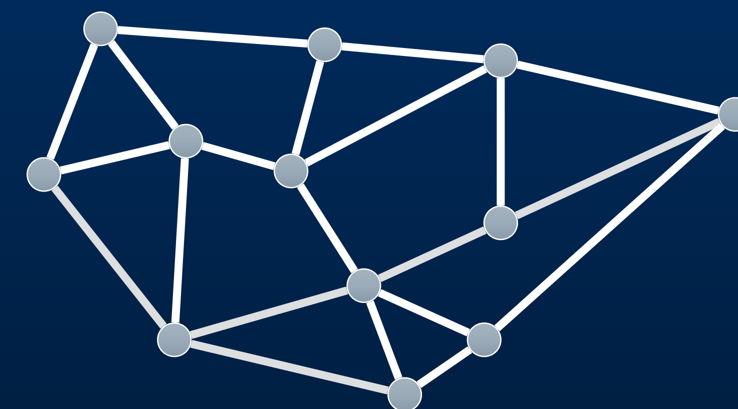
$$(n-1) \cdot MF(n) = \tilde{O}(n^3)$$

Both $\tilde{O}(n^3)$ but APSP seems easier.

All-Pairs MF vs. All-Pairs SP

Unweighted
(simple graphs)





Gomory-Hu 1961:

$$(n-1)\cdot MF(n) = \tilde{O}(n^3)$$

[Bhalgat-Hariharan-Kavitha-Panigrahi '07]

$\tilde{O}(mn)$ *(without MF)*

Siedel 1995:

$$O(n^{\omega}) = O(n^{2.37})$$

APSP is subcubic, APMF is not: a separation, finally?

APMF vs. APSP

	APMF	APSP
General	n^3	n^3
Unweighted	n^3	$n^{2.37}$

It feels like **APMF** $\geq$ **APSP***…*

Is it so?

Enter:
Fine-Grained Complexity

A small set of "conjectures"

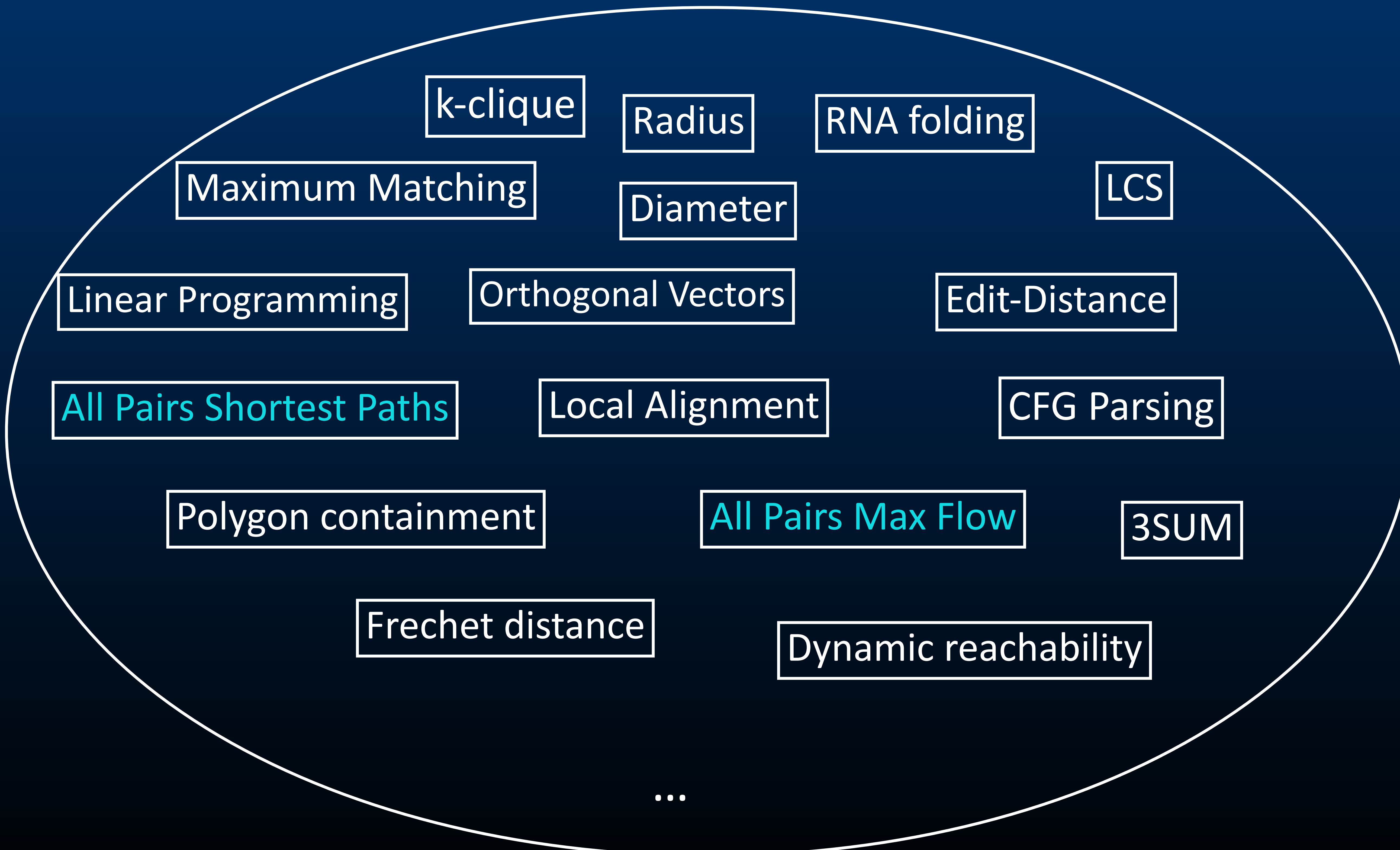
(APSP, 3SUM, SETH, ...)

↓ *fine-grained reductions*

Tight lower bounds for
lots of natural and important problems

$\Omega(n^3), \Omega(n^{2.5}), \Omega(n^2), \ldots$

The Class P (before)



The Class P (after)

k-SAT

- Diameter
- Closest Pair
- Local Alignment
- Dynamic Reachability
- Single-Source Max-Flow
- Subtree Isomorphism
- Stable Matching
- Edit-Distance
- Frechet
- LCS
- …

3SUM

- Colinearity
- Polygon Containment
- Strips Cover Rectangle
- Triangle Enumeration
- Compressed Inner Product
- Dynamic Max Matching
- Set Intersection
- …

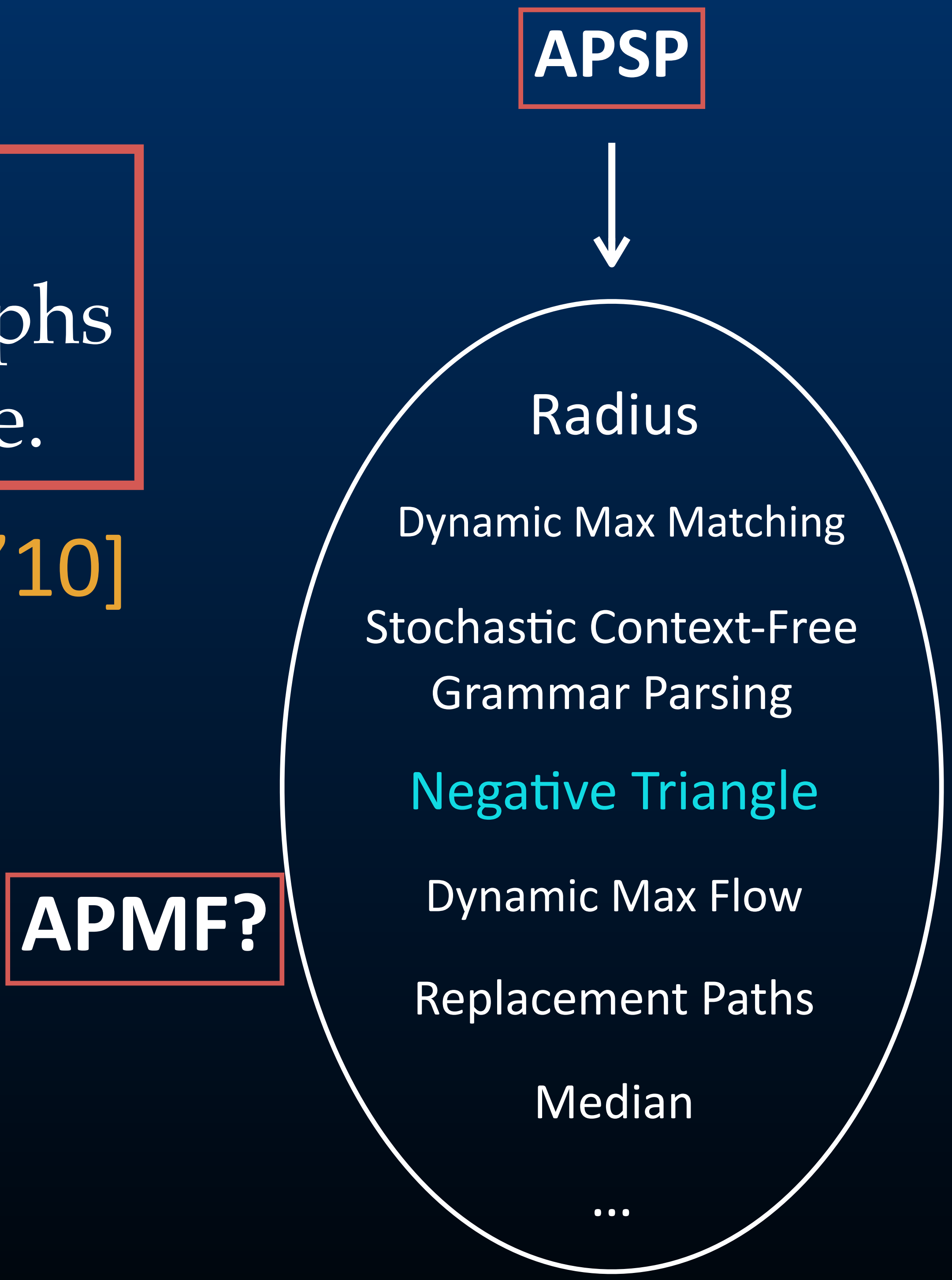
APSP

- Radius
- Dynamic Max Matching
- Stochastic Context-Free Grammar Parsing
- Negative Triangle
- Dynamic Max Flow
- Replacement Paths
- Median
- …

APSP-Hard Problems

Conjecture:
APSP in undirected weighted graphs cannot be solved in $O(n^{3-\varepsilon})$ time.

[Vassilevska Williams - Williams '10]



It feels like **APMF** $\geq$ **APSP***...*
Is it so?

APMF > APSP in directed graphs

Assuming **SETH**:

$\Omega(n^{2-\varepsilon})$ for Single-Source MF in sparse graphs [A.- Vassilevska W. - Yu ’15]

vs. $\tilde{O}(n)$ for Single-Source SP in sparse graphs

$\Omega(n^{3-\varepsilon})$ for **APMF** in sparse graphs [Krauthgamer-Trablesi ’17]

vs. $\tilde{O}(n^2)$ for APSP in sparse graphs

Assuming the **4-Clique** Conjecture:

$\Omega(n^{\omega+1-\varepsilon}) = \Omega(n^{3.37})$ for **APMF** in dense graphs [**A**GIKPTUW ’19]

vs. $\tilde{O}(n^3)$ for APSP in dense graphs

APMF is indeed harder than APSP… in directed graphs.

APMF > APSP in directed graphs

Assuming **SETH**:

$\Omega(n^{2-\varepsilon})$ for Single-Source MF in sparse graphs [A.- Vassilevska W. - Yu '15]

vs. $\tilde{O}(n)$ for Single-Source SP in sparse graphs

$\Omega(n^{3-\varepsilon})$ for **APMF** in sparse graphs [Krauthgamer-Trablesi '17]

vs. $\tilde{O}(n^2)$ for APSP in sparse graphs

Assuming the **4-Clique** Conjecture:

$\Omega(n^{\omega+1-\varepsilon}) = \Omega(n^{3.37})$ for **APMF** in dense graphs [**A**GIKPTUW '19]

vs. $\tilde{O}(n^3)$ for APSP in dense graphs

[A.- Krauthgamer - Trabelsi SODA'20]:
Same lower bounds for undirected graphs with *node capacities.*

APMF > APSP ?

APMF is indeed harder than APSP…

Undirected Graphs	APMF	APSP
General	n^3	n^3
Unweighted	n^3	$n^{2.37}$

…But only where the Gomory-Hu result does not apply!

Other settings	APMF	APSP
Directed	$\Omega(n^{3.37}), O(n^4)$	n^3
Node-capacities	$\Omega(n^{3.37}), O(n^4)$	n^3

[Mayeda'62,Jelinek'63,Hassin-Levine'07]

GH Trees are impossible in these settings.

(Because there are $\Omega(n^2)$ answers.)

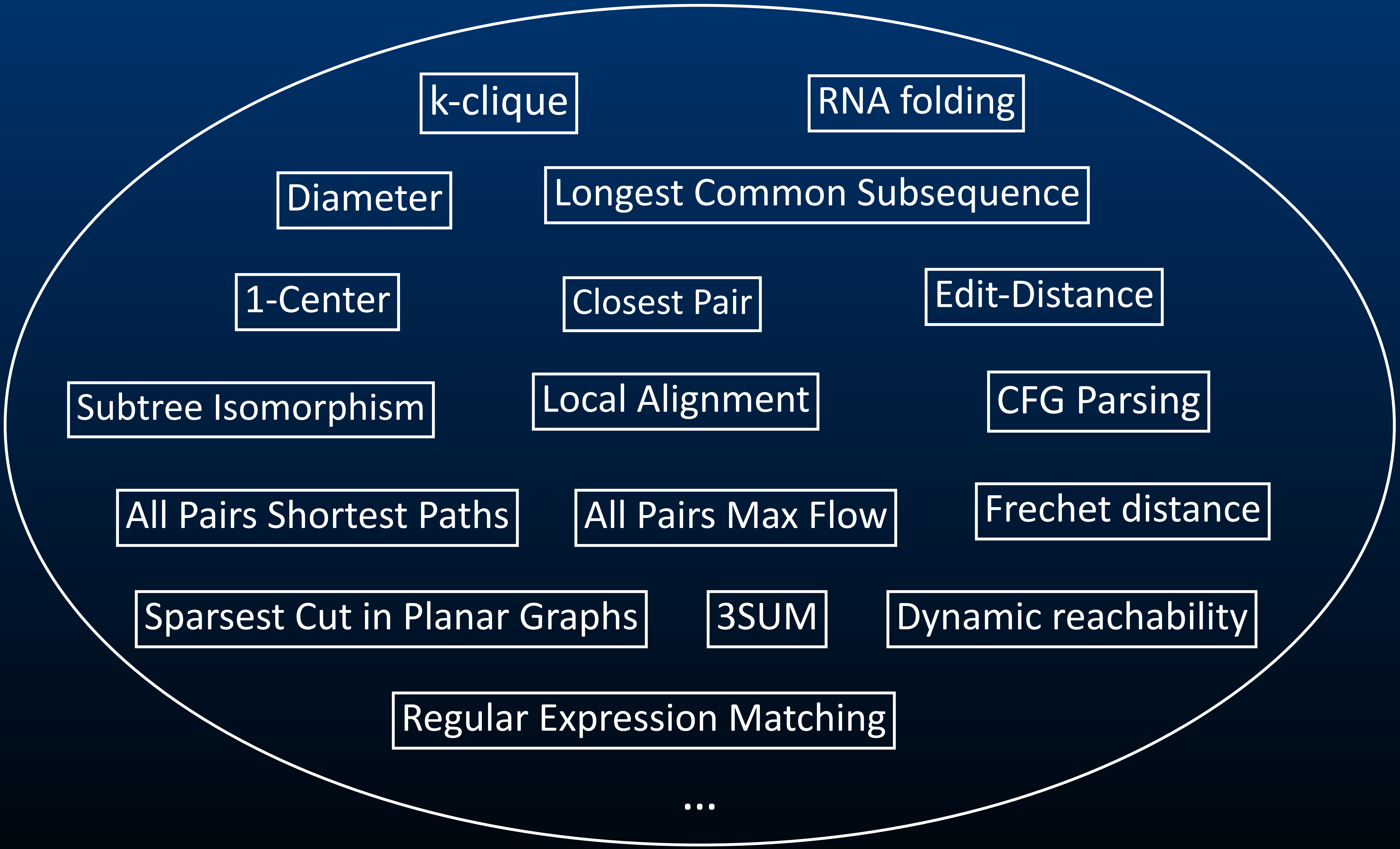
APMF > APSP ?

Undirected Graphs	APMF	APSP
General	n^3	n^3
Unweighted	n^3	$n^{2.37}$

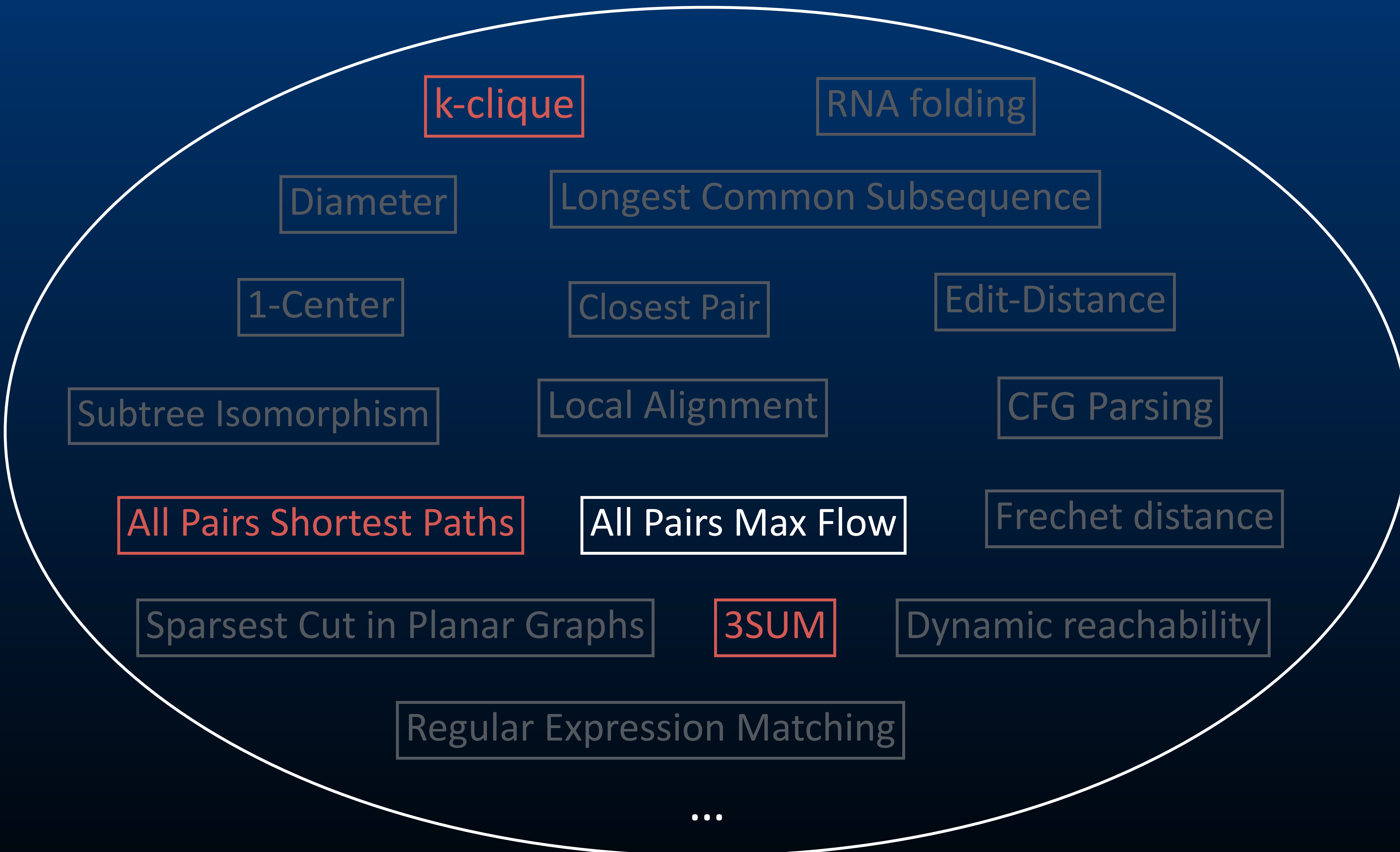
Open Question:

Prove any “fine-grained complexity” lower bounds for APMF.

Me 10-years ago: *"why work on APMF vs. any other problem?"*



<u>Fine-Grained Complexity:</u> *"APMF is really worth studying…"*



Non-reducibility?

Open Question:
Prove any "fine-grained complexity" lower bounds for APMF.

[Carmosino - Gao - Impagliazzo - Mikhailin - Paturi - Schneider '16]
"No reduction between problems with different non-deterministic complexity (assuming NSETH)"

Are we failing to prove hardness because APMF is easy nondeterministically?

[A.- Krauthgamer - Trabelsi SODA'20]:
There is a prover-verifier protocol for constructing a GH-Tree of an unweighted graph, where the verifier only takes $\tilde{O}(m)$ time.

Nondeterministic Algorithms

Real algorithms	APMF	APSP
General	n^3	n^3
Unweighted	n^3	$n^{2.37}$

Nondeterministic algorithms	APMF	APSP
General	n^3	$n^{2.94}$ [CGIMPS '16]
Unweighted	n^2 [AKT '20]	$n^{2.37}$

Did we really need the nondeterminism?

Towards Faster APMF

Did we really need the nondeterminism?

[A.- Krauthgamer - Trabelsi FOCS'20]:

1. **GHT** and **APMF** reduce to ***Single-Source*** Max-Flow. * *SS values suffice.*

2. $(1+\varepsilon)$-**APMF** in $\tilde{O}(n^2)$ time. * *Also* $(1+\varepsilon)$-**GHT** *in* $\tilde{O}(MF(n))$.

3. "Cut-Equivalent Trees are Optimal for Min-Cut Queries"

* [Li-Panigrahi STOC'21]

APMF vs. APSP

Exact	APMF	APSP
General	n^3	n^3
Unweighted	n^3	$n^{2.37}$

Approximation	APMF	APSP
3		n^2 [Dor-Halperin-Zwick'00]
$poly \log n$	n^2 [Racke'04]	
$(1+\varepsilon)$	n^2 [AKT '20,LP'21]	$n^{2.37}$ [Zwick'02]

Towards Faster APMF

All we have to do is solve Single-Source Max-Flow…

[A.- Krauthgamer - Trabelsi FOCS'20]:

1. **GHT** and **APMF** reduce to ***Single-Source*** Max-Flow. * *SS values suffice.*

2. $(1+\varepsilon)$-**APMF** in $\tilde{O}(n^2)$ time. * *Also* $(1+\varepsilon)$-**GHT** *in* $\tilde{O}(MF(n))$.

3. "Cut-Equivalent Trees are Optimal for Min-Cut Queries"

* [Li-Panigrahi STOC'21]

Breaking the Cubic Barrier

[A.- Krauthgamer - Trabelsi STOC'21]:

GH Tree in $\tilde{O}(n^{2.5})$ time for simple graphs.

	APMF	**APSP**
General	n^3	n^3
Unweighted	$n^{2.5}$ [AKT '21]	$n^{2.37}$ [Siedel '95]

<u>Main tools:</u>
- *Expander-decomposition*
- "Isolating Cuts"

Still APMF > APSP but we are not so sure anymore…

APMF < APSP for simple graphs?

[A.- Krauthgamer - Trabelsi FOCS'21, Li-Panigrahi-Saranurak FOCS'21]
(see also [Zhang '21]):
GH Tree in $n^{2+o(1)}$ time for simple graphs.

	APMF	**APSP**
General	n^3	n^3
Unweighted	n^2	$n^{2.37}$ [Siedel '95]

[**A**KT '21, LPS'21, Zhang'21]

Main tool:
Expander-decomposition with vertex demands

- ✓ Optimal for APMF.
- ✓ All-Pairs in single-pair time!
- ✓ Can be derandomized [**A**KT'21]

GH Tree in Linear Time?

Still open: **GH Tree in $\tilde{O}$(*Max-Flow-Time*) ?**

$$n^{2+o(1)} \rightarrow m^{1+o(1)}$$

So far only known for bounded genus, and bounded tree width graphs.

[A.- Krauthgamer - Trabelsi FOCS'21]

$\Omega(m + n^{1.5})$ for **GH Tree** in simple graphs,

assuming $\Omega(n^3)$ for multigraphs.

[A.- Krauthgamer - Trabelsi SODA'22]

$(m + n^{1.9})^{1+o(1)}$ for **GH Tree** in simple graphs,

via new "Friendly Cut Sparsifiers".

Main Open Question:

APMF in subcubic time for general graphs?

APMF < APSP ?

No separation when $\omega = 2$...

	APMF	APSP
General	n^3	n^3
Unweighted	n^2	n^ω [Siedel '95]

[AKT '21, LPS'21, Zhang'21]

<u>Main Open Question:</u>
APMF in subcubic time for general graphs?

Would imply a true separation under the APSP Conjecture.

Gomory-Hu's 1961 algorithm remains the only solution for the general case...

Even nondeterministically.

APMF in Subcubic Time

[A.- Krauthgamer - Li - Panigrahi - Saranurak - Trabelsi (new!)]:

GH Tree in $\tilde{O}(n^{2.875})$ time.

Exact	APMF	APSP
General	$n^{2+7/8}$	n^3
Unweighted	n^2	n^ω

Assuming the APSP Conjecture: **APMF < APSP!**

So maybe the APSP Conjecture is false…?

APMF in Quadratic Time!

21 days later…

[A.- Krauthgamer - Li - Panigrahi - Saranurak - Trabelsi (v2), and independently Zhang'21]:

GH Tree in $\tilde{O}(n^2)$ time.

Exact	APMF	APSP
General	n^2	n^3
Unweighted	n^2	n^ω

Assuming (a very weak) APSP Conjecture: **APMF < APSP!**

Gomory-Hu Tree in Max-Flow Time?

[A.- Krauthgamer - Li - Panigrahi - Saranurak - Trabelsi (v2), and independently Zhang'21]:

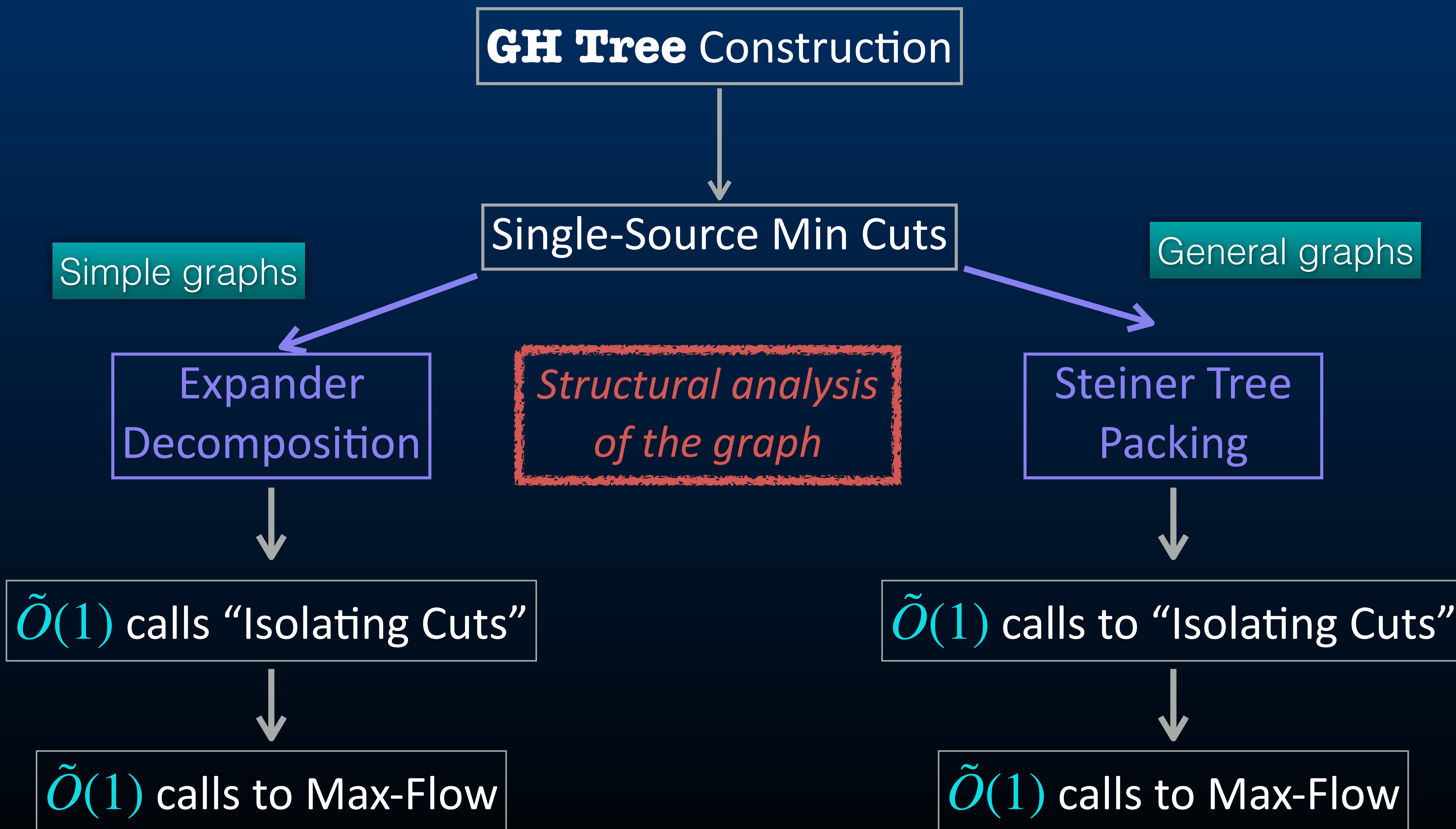
GH Tree in time:

- $\tilde{O}(n^2) + \tilde{O}(\text{Max-Flow}(n, m))$ for weighted graphs,
- $m^{1+o(1)} + \tilde{O}(\text{Max-Flow}(n, m))$ for unweighted graphs.

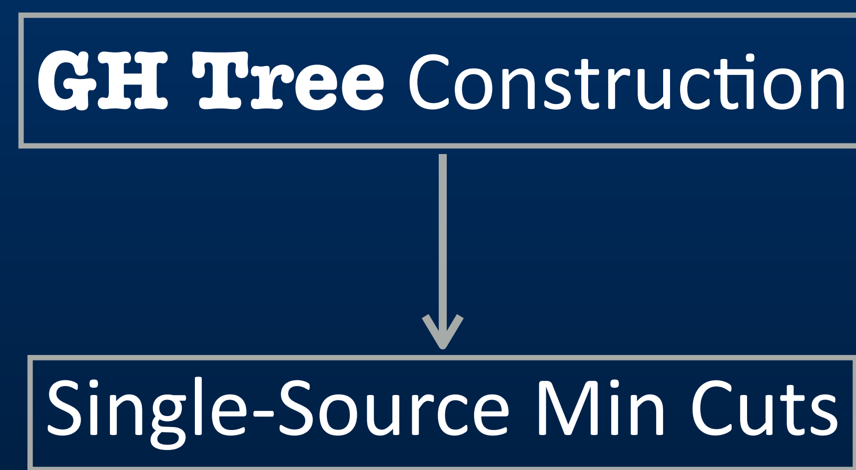
*We can compute a succinct **all-pairs** max-flow oracle in the time to compute a **single-pair** max-flow!*

Technical Overview

How to construct a GH Tree in subcubic time

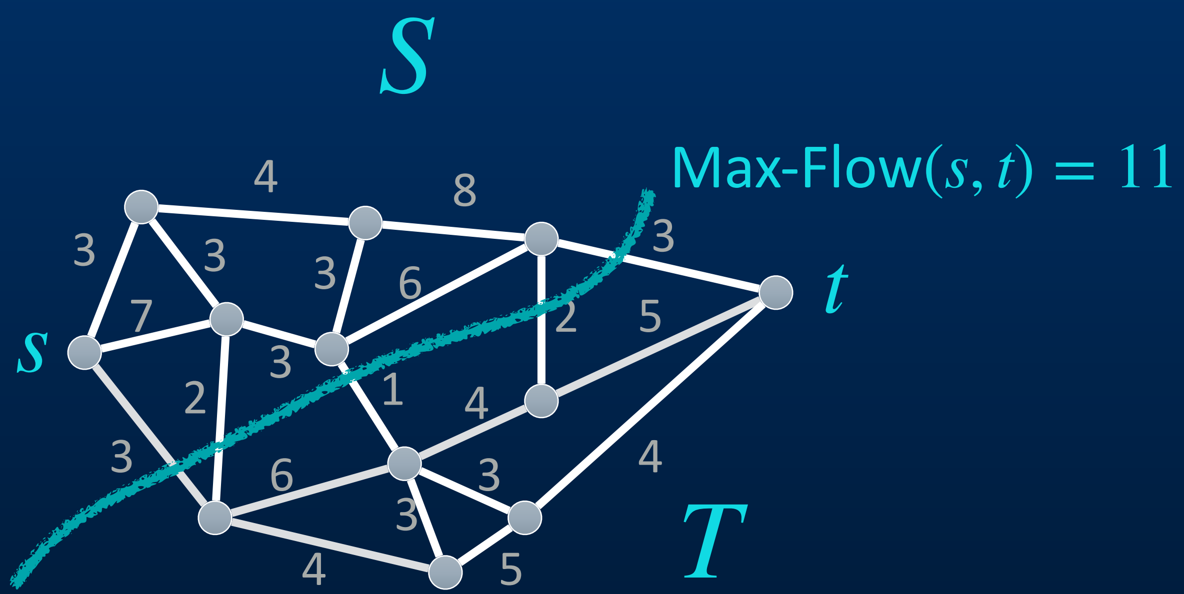


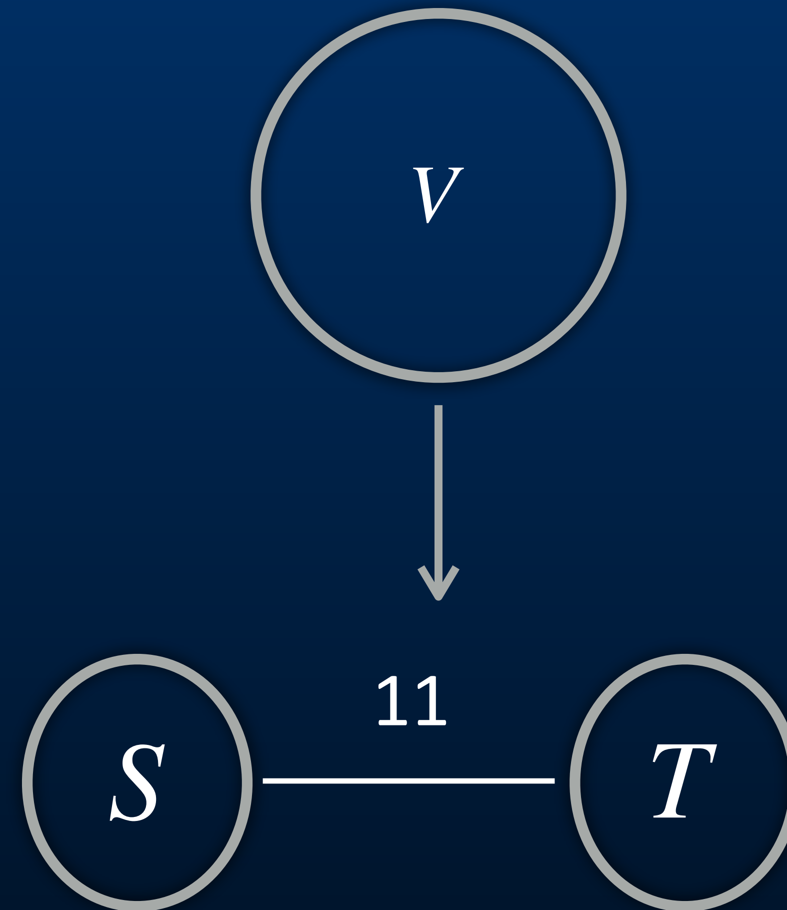
Technical Overview



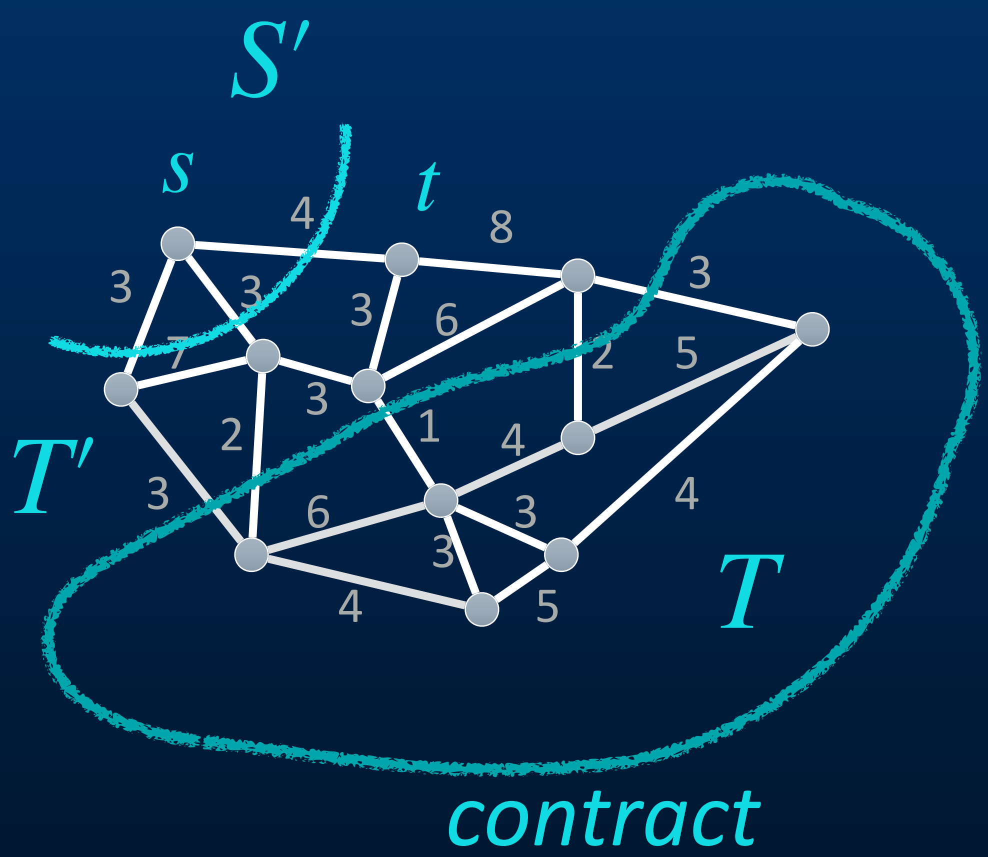
Let's start with the GH algorithm
and optimize it with single-source cuts

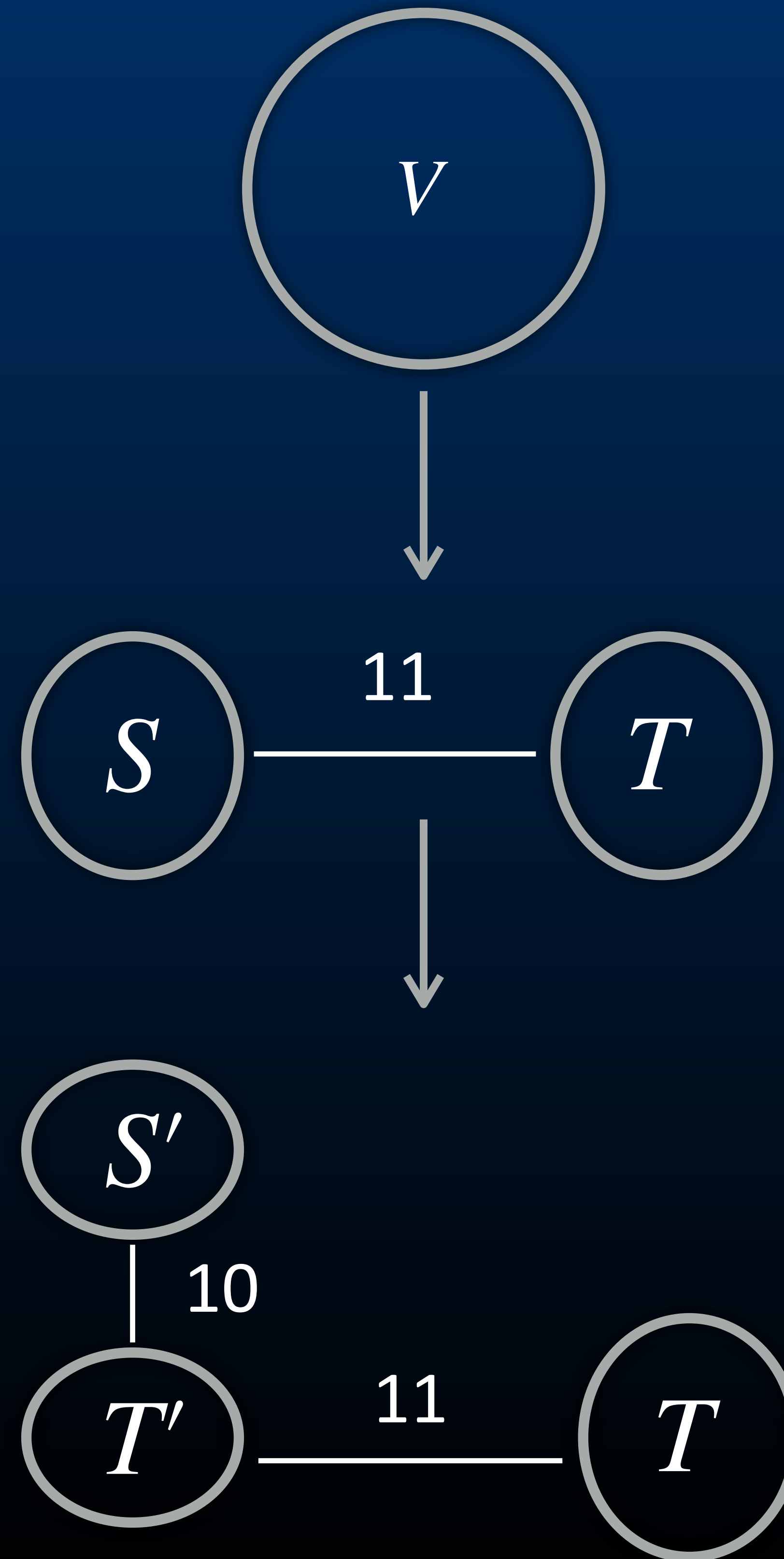
The Gomory-Hu Algorithm





The Gomory-Hu Algorithm

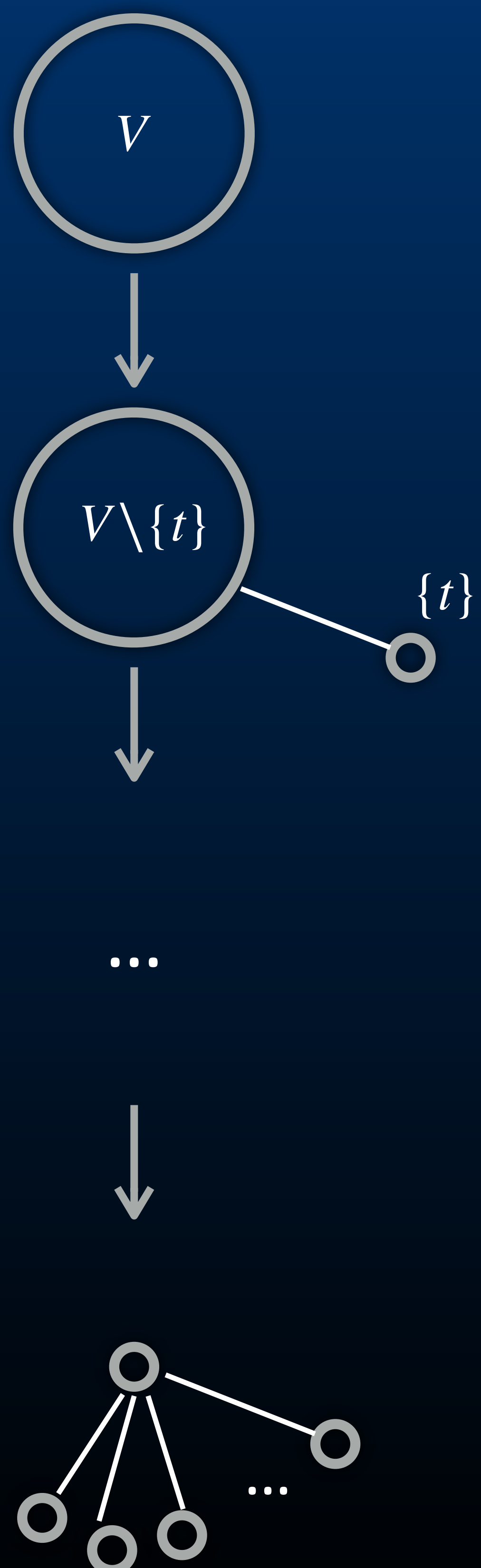




Lemma [Gomory-Hu]:

The min s,t-cut in the contracted graph is a min s,t-cut in the original graph.

The Gomory-Hu Algorithm

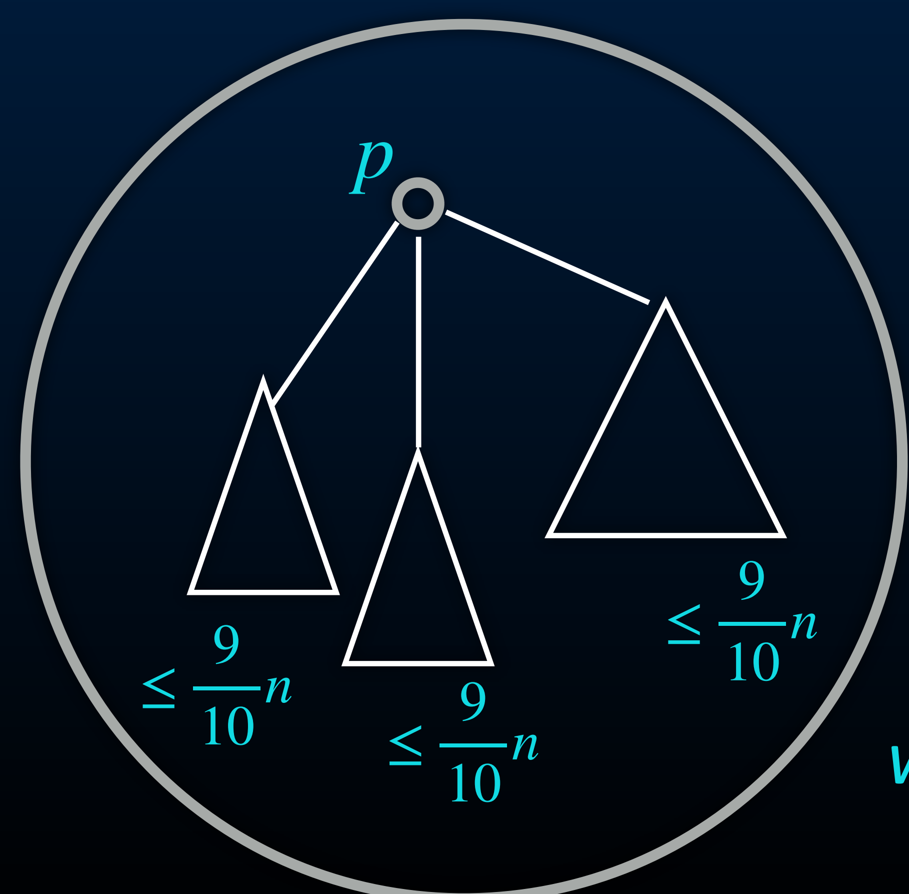


Always makes n-1 calls to Max-Flow,
but the graphs get smaller…

Worst case: Recursion depth $\Omega(n)$.

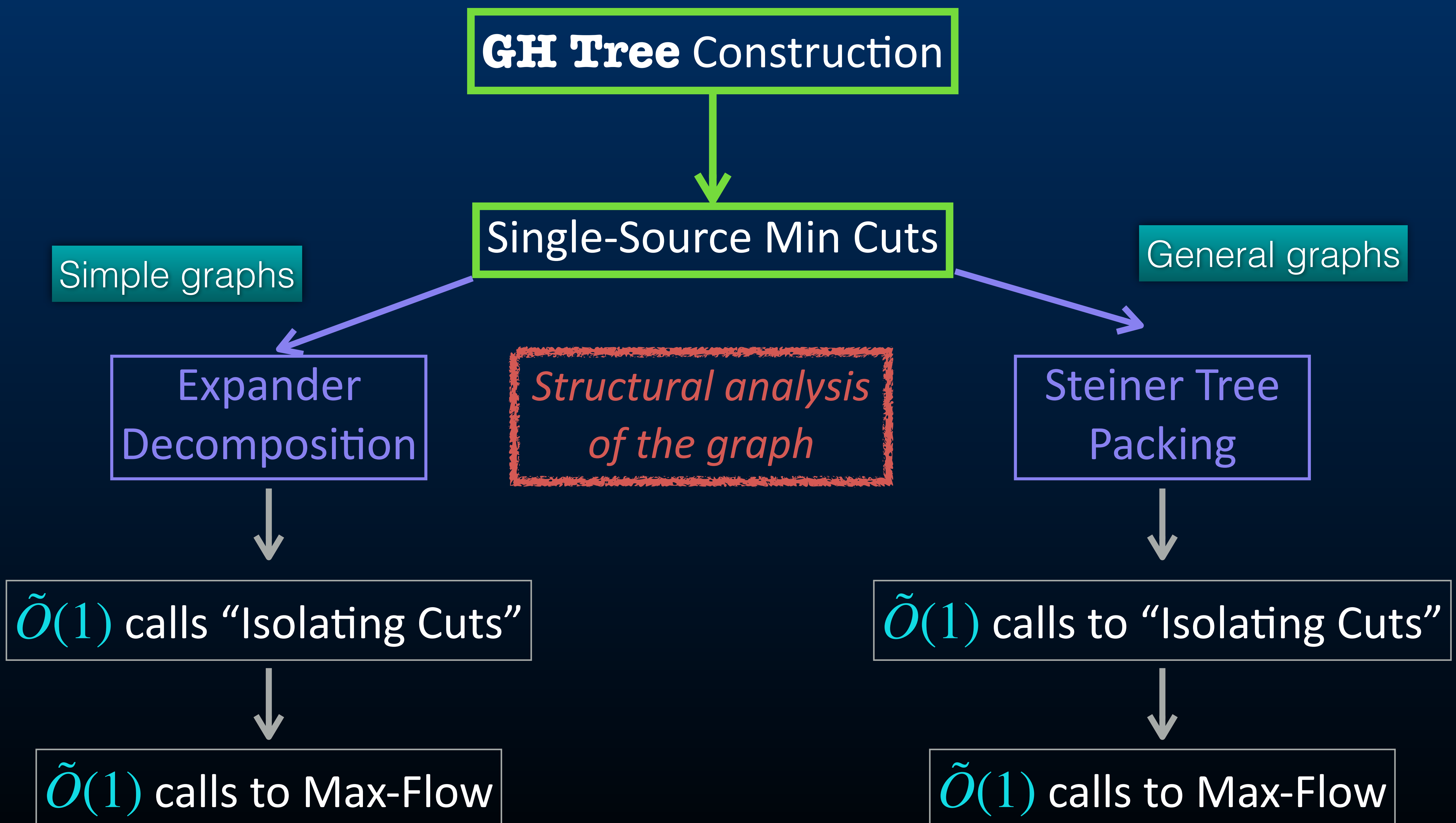
$$\Omega(n) \cdot MF(n)$$

Speedup idea: Use single-source cuts,
from a random pivot.



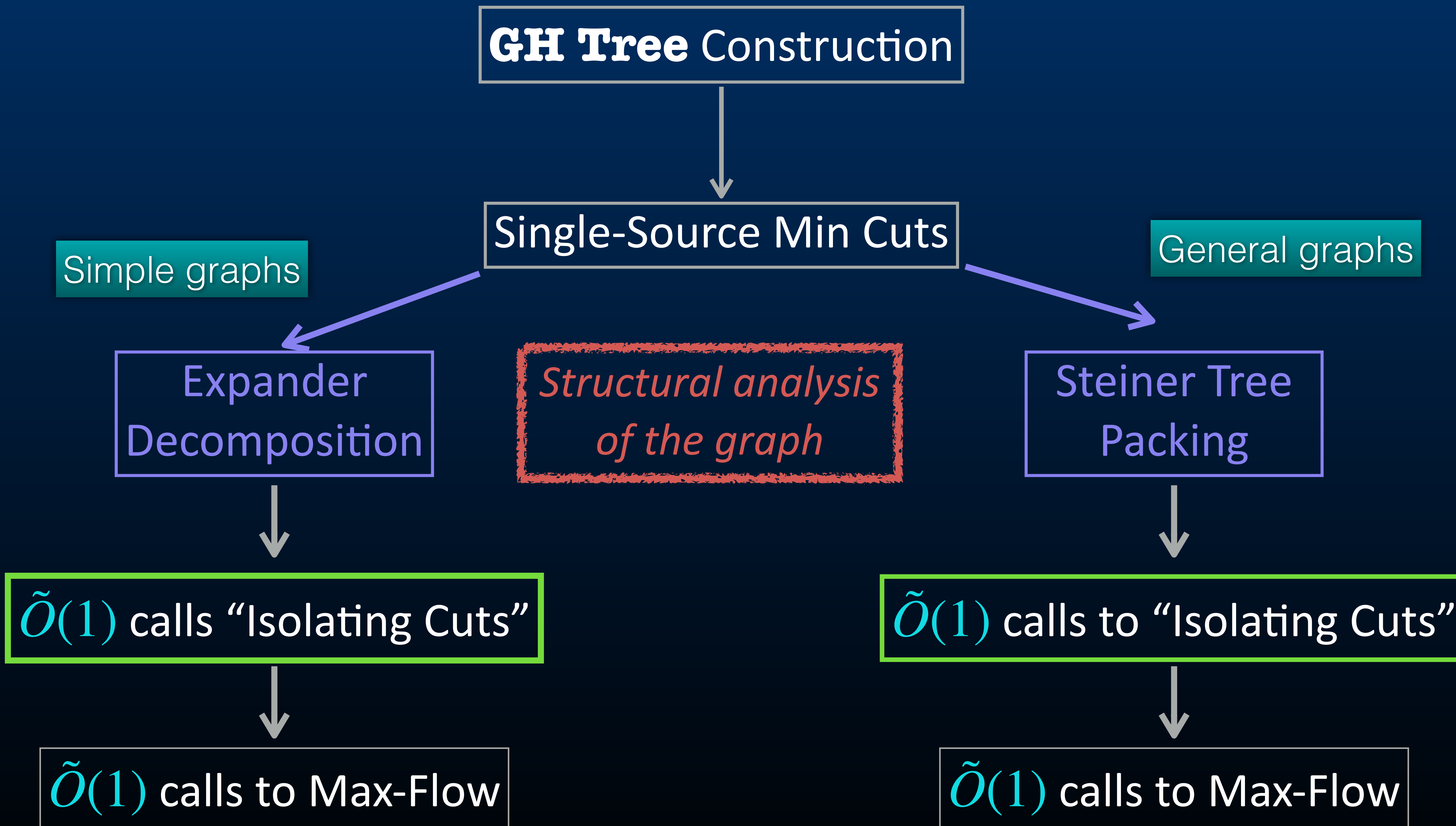
w.h.p

Technical Overview



The Isolating Cuts Procedure

*Discovered independently by [LP'20,**AKT'21**]*

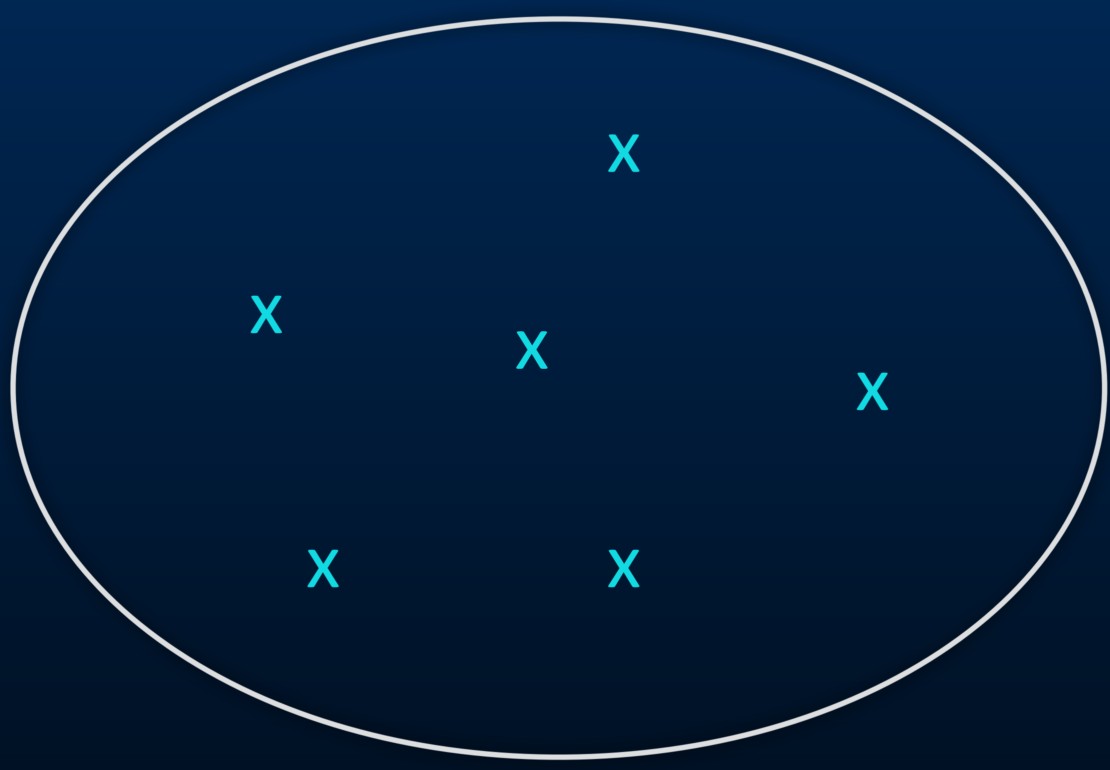


The Isolating Cuts Procedure

*Discovered independently by [LP’20,**A**KT’21]*

and quickly found many applications

*[LP21, CQ21, MN21, LNPSY21, **A**KT21a, LPS21, Zha21, **A**KT22, CLP22]*



Given a set of terminals $U = \{u_1, \ldots, u_k\}$

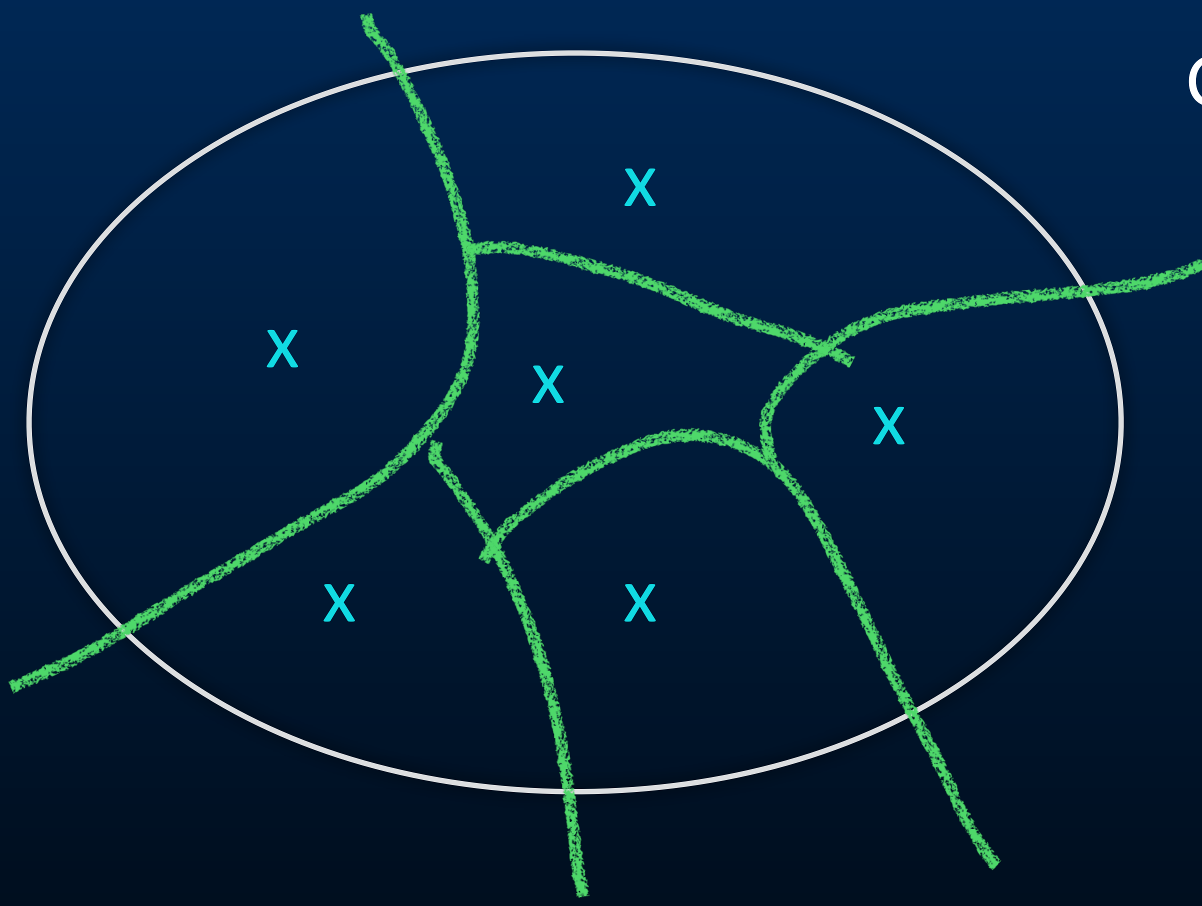
return $\forall u_i \in U$: min $(u_i, U \setminus \{u_i\})$-cut

The Isolating Cuts Procedure

*Discovered independently by [LP'20,**A**KT'21]*

and quickly found many applications

*[LP21, CQ21, MN21, LNPSY21, **A**KT21a, LPS21, Zha21, **A**KT22, CLP22]*



Given a set of terminals $U = \{u_1, \ldots, u_k\}$

return $\forall u_i \in U$: min $(u_i, U \setminus \{u_i\})$-cut

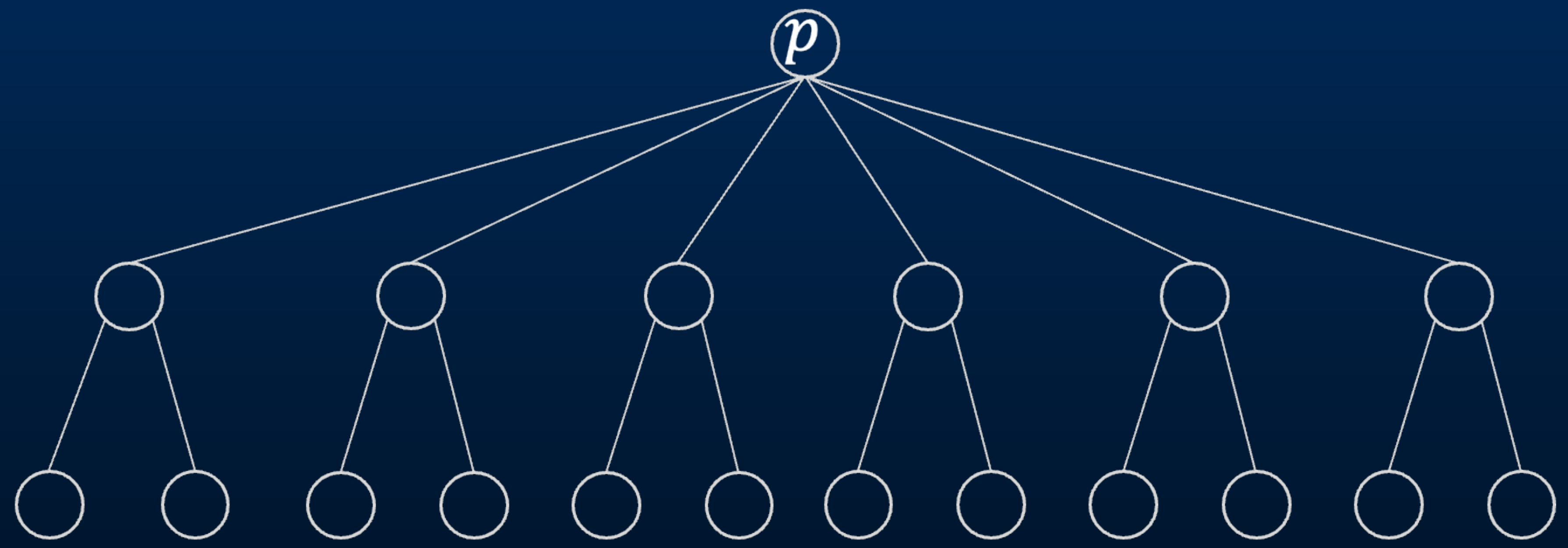
"Isolating cuts"

Only takes $\tilde{O}(MF(n,m))$ time.

Not $k \cdot MF(n,m)$...

SS Min-Cuts via Isolating Cuts

A hard case: suppose that this is the GHT.



Challenge: Find the triples (well-connected triangles).

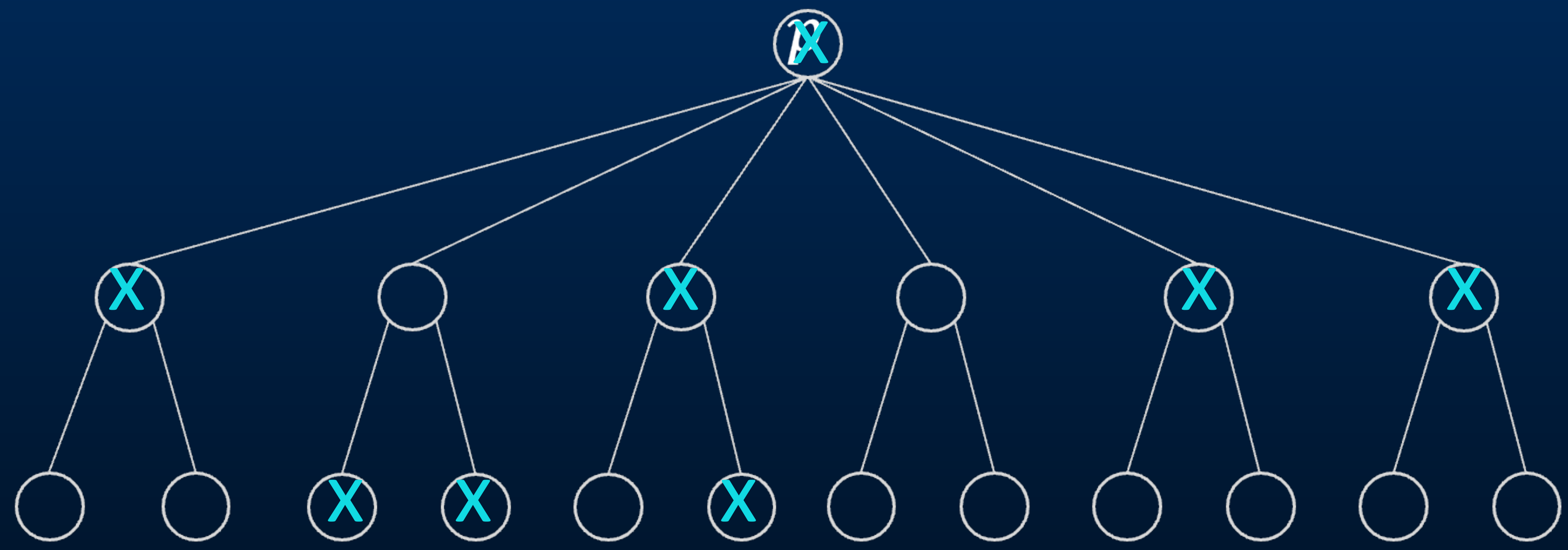
Reminds of the max-triangle problem (APSP-equivalent)…

Solution: Use Isolating Cuts with random terminals.

Figure by Ohad Trabelsi.

SS Min-Cuts via Isolating Cuts

A hard case: suppose that this is the GHT.

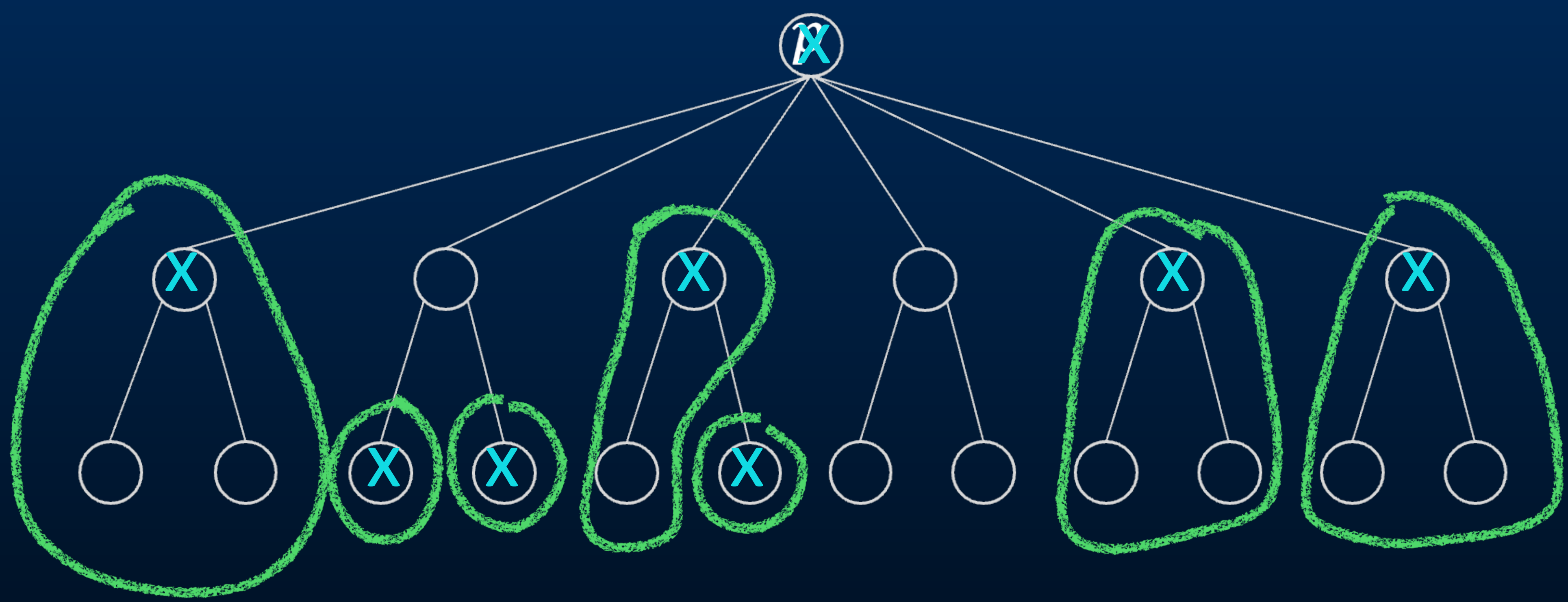


Challenge: Find the triples (well-connected triangles).

Solution: Use Isolating Cuts with random terminals.

SS Min-Cuts via Isolating Cuts

A hard case: suppose that this is the GHT.



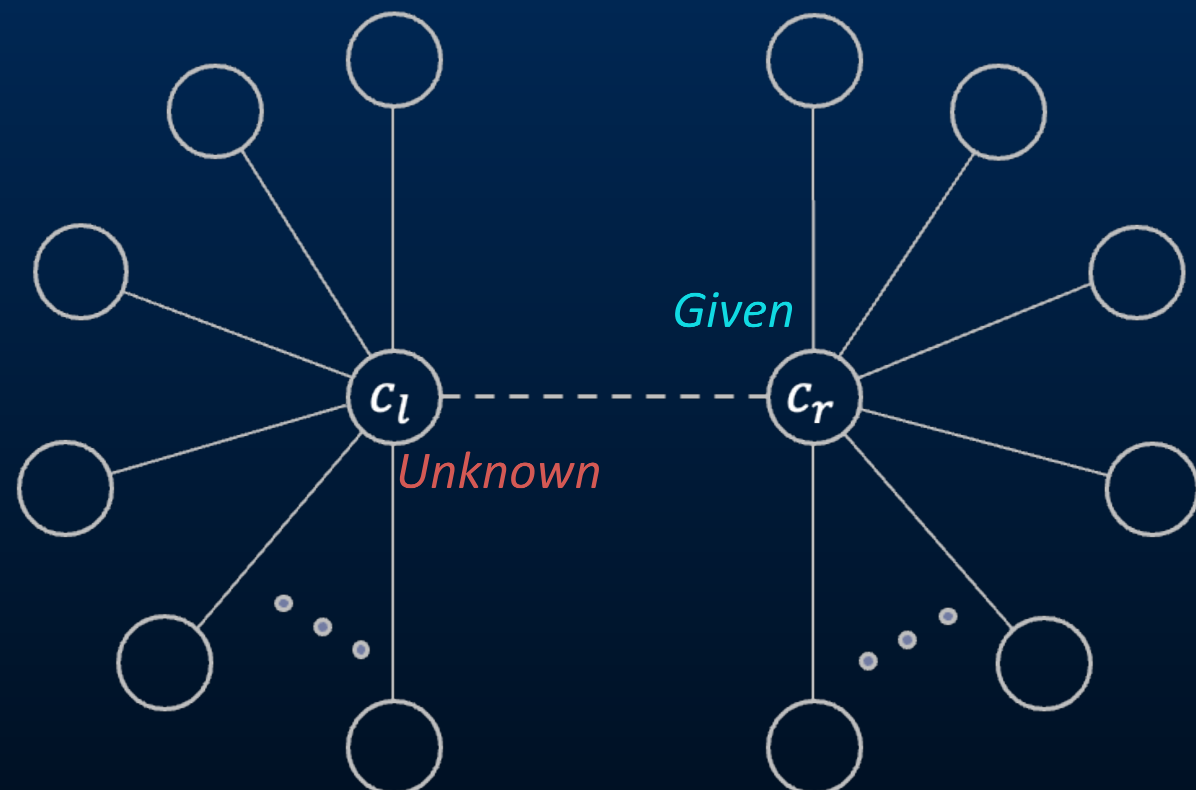
Challenge: Find the triples (well-connected triangles).

Works well for finding unbalanced cuts…

Solution: Use Isolating Cuts with random terminals.

SS Min-Cuts via Isolating Cuts

The hardest case: suppose that this is the GHT.



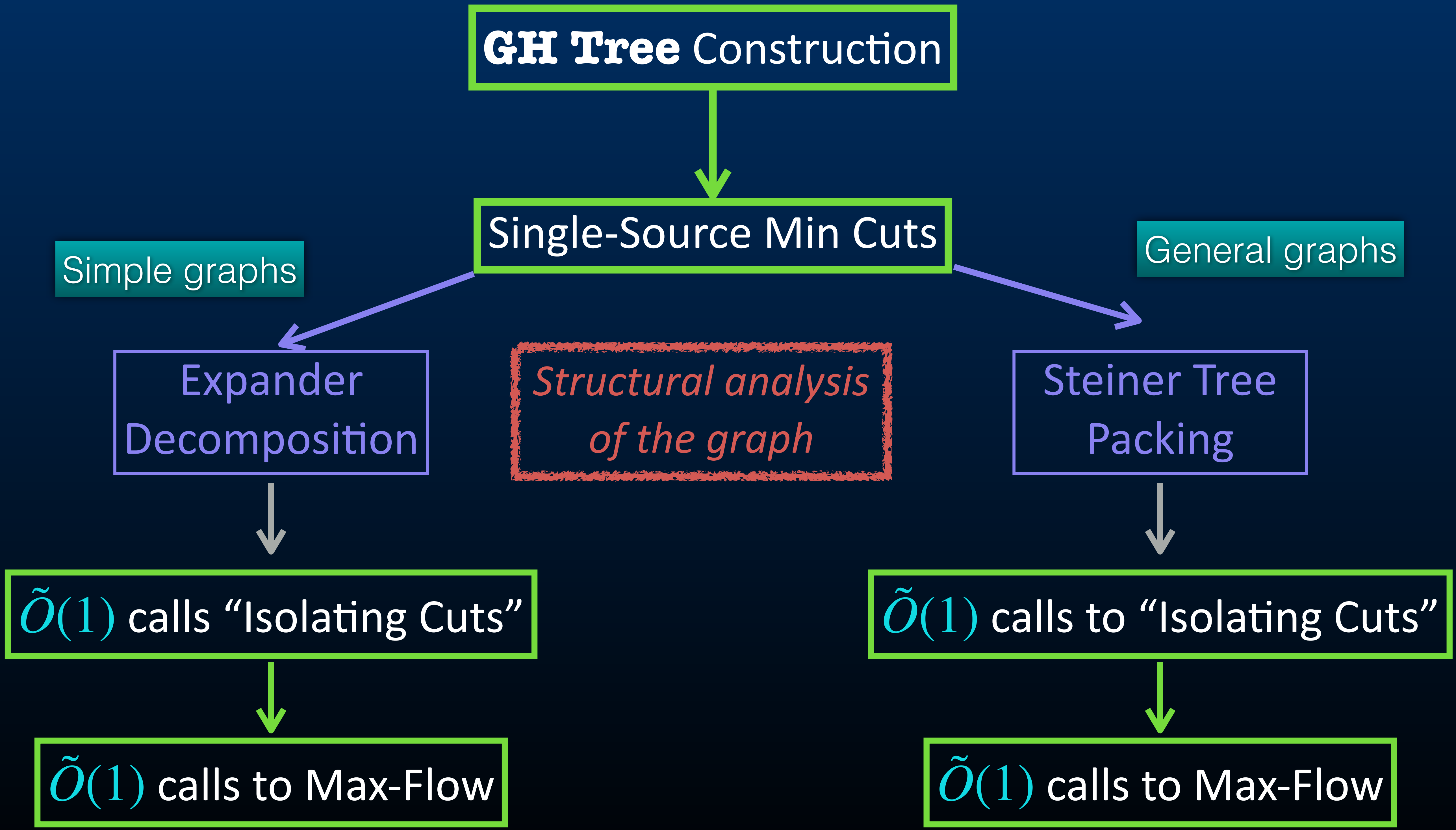
One non-trivial large cut; how to find it?

Challenge: The min (c_r, v)-cuts tell us nothing about c_l.

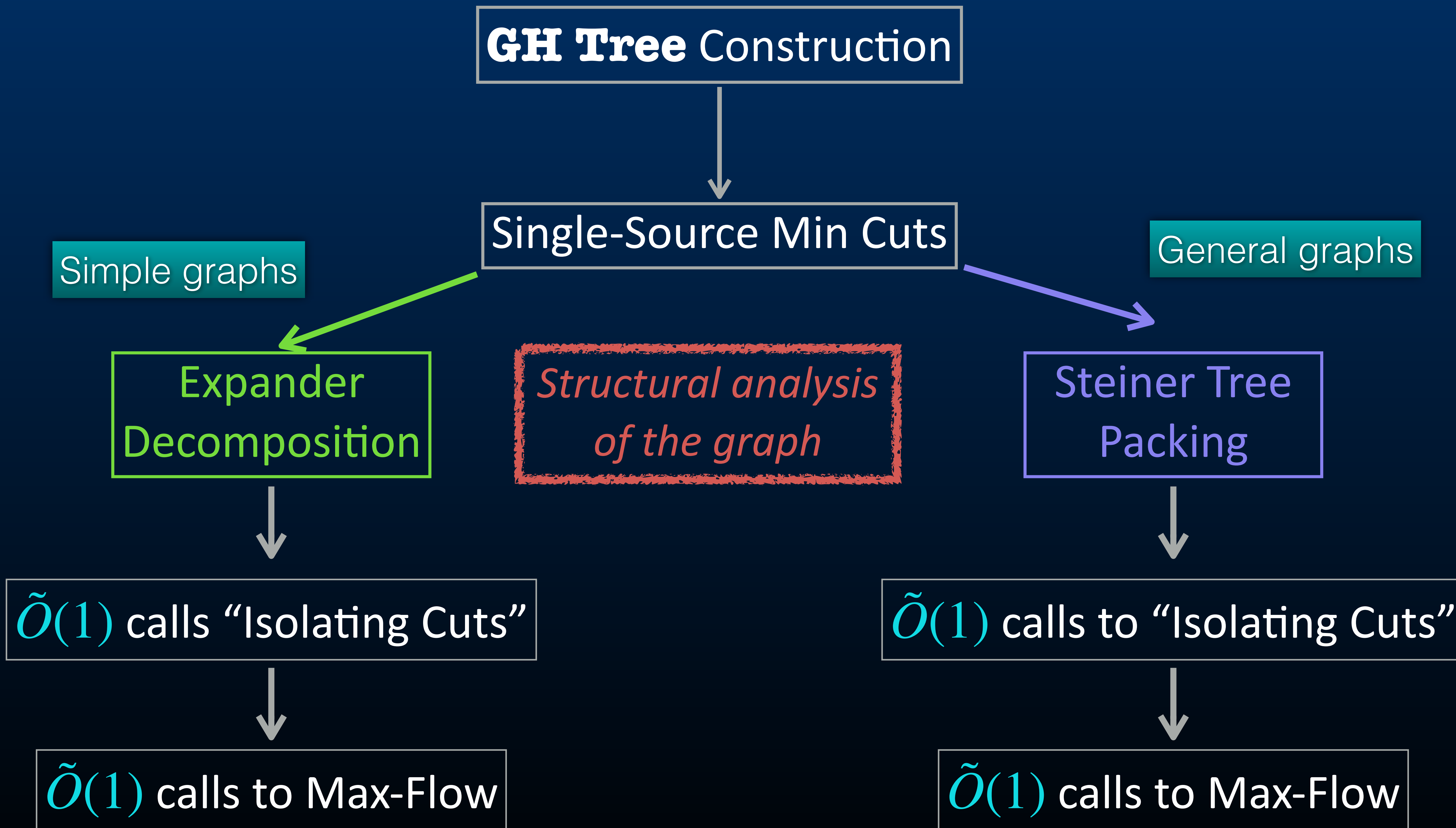
Solution: Use a structural analysis of the graph as a guide.

Figure by Ohad Trabelsi.

Technical Overview



Expanders-guided querying



Expanders-guided querying

An expander decomposition of G:

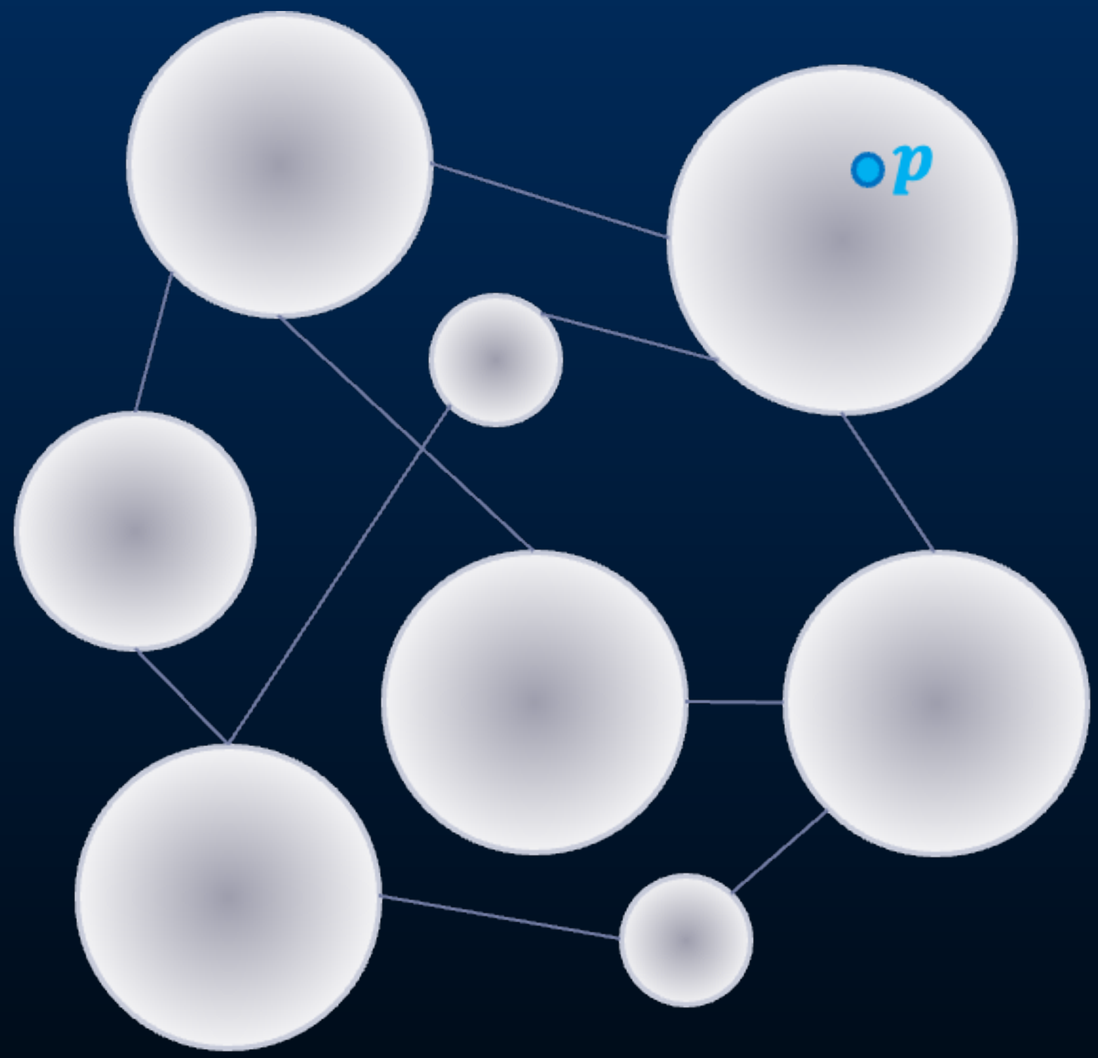
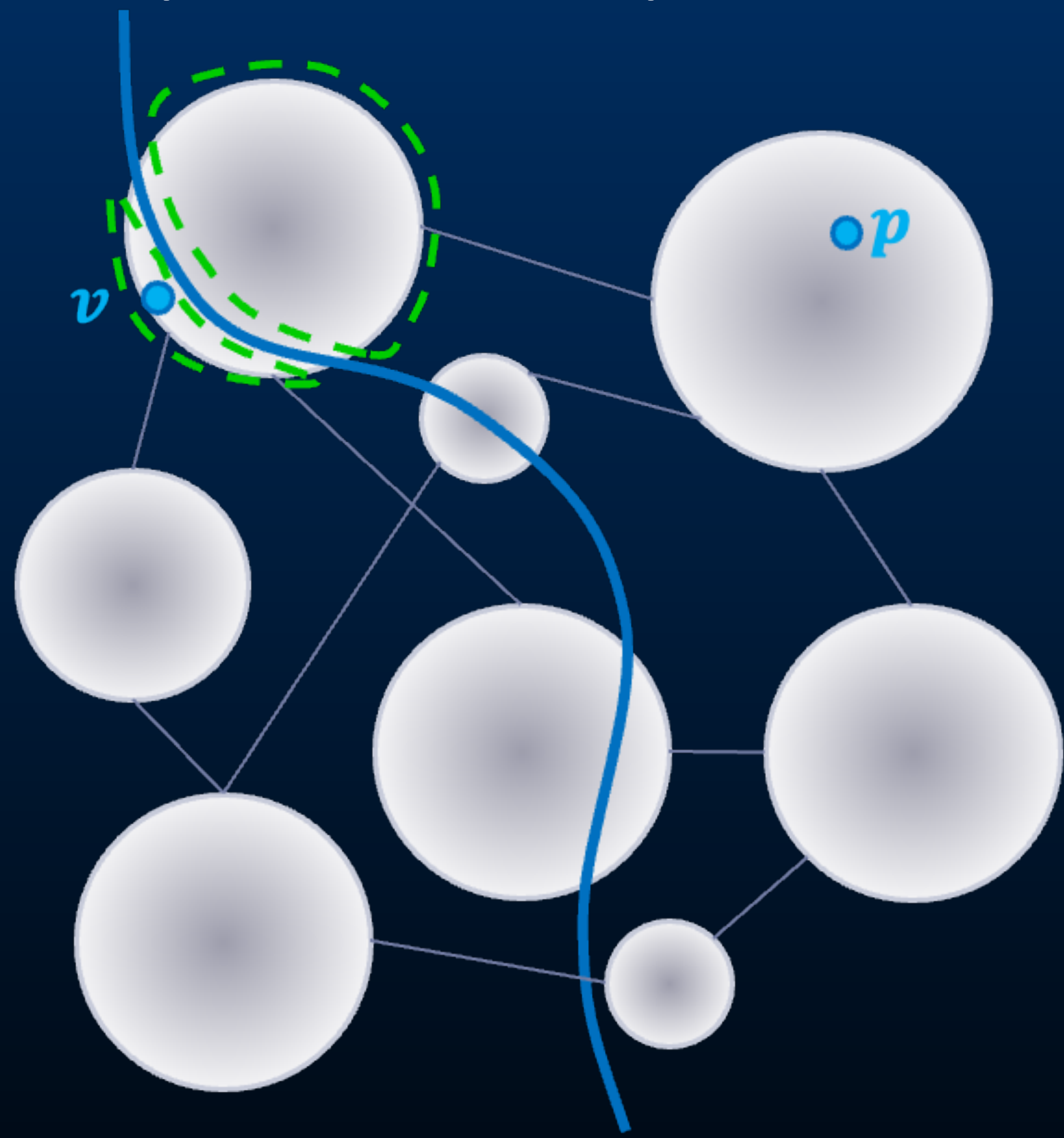


Figure by Ohad Trabelsi.

Expanders-guided querying

An expander decomposition of G:

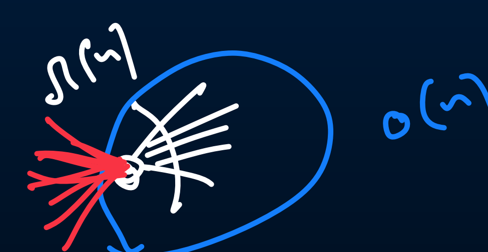


$\tilde{O}(n^{2.5})$ for simple graphs: [AKT STOC'21]

Set: $\phi = 1/\sqrt{n}$

1. In any expander:
   $\leq \sqrt{n}$ nodes from one side

2. $\leq n^{1.5}$ edges outside expanders.



$O(1)$ expanders with $\Omega(n)$ nodes.

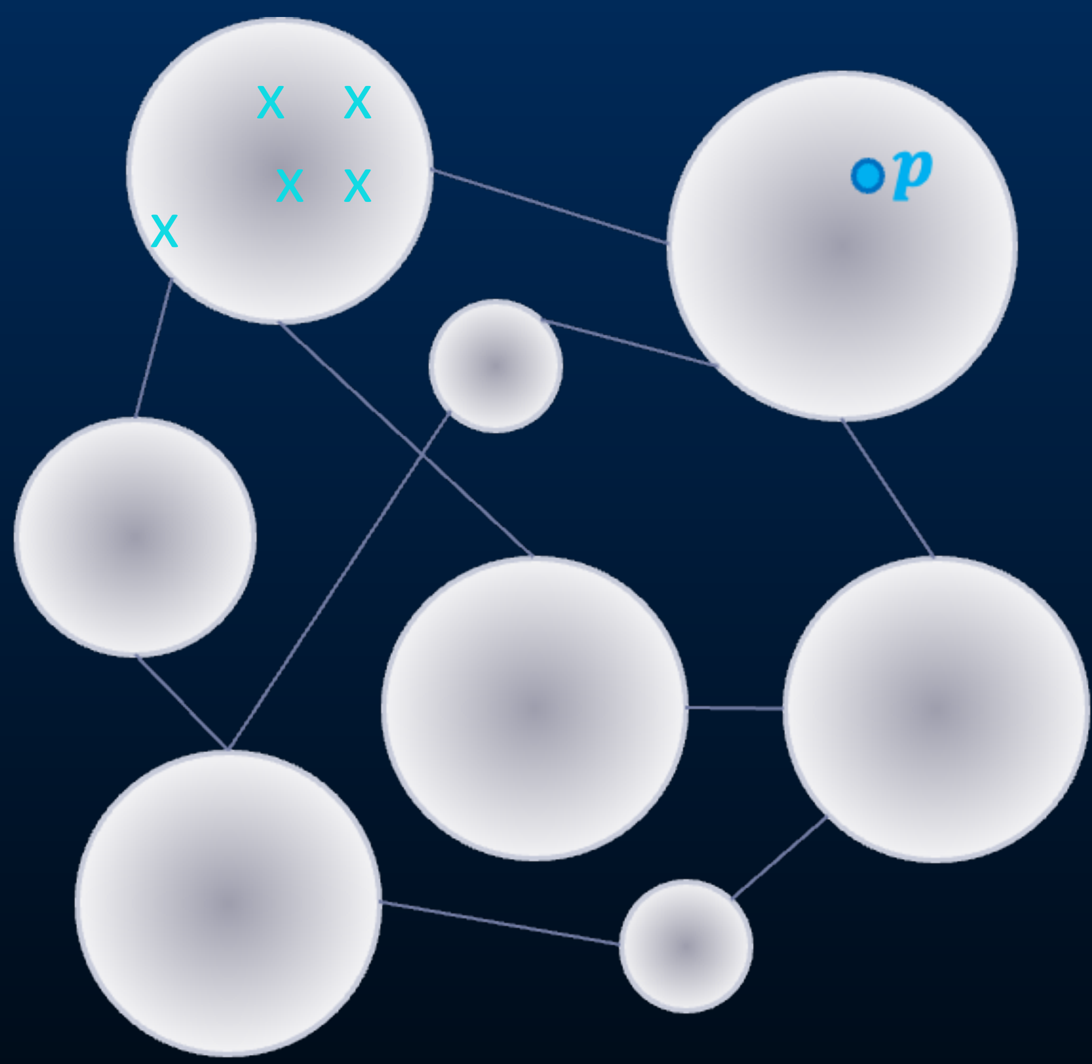
Simple graphs only:

$\leq \sqrt{n}$ nodes in small expanders.

Figure by Ohad Trabelsi.

Expanders-guided querying

An expander decomposition of G:



For each large expander:
Use Isolating Cuts with random terminals.

$\tilde{O}(n^{2.5})$ for simple graphs: [AKT STOC'21]

Set: $\phi = 1/\sqrt{n}$

1. In any expander:
   $\leq \sqrt{n}$ nodes from one side

2. $\leq n^{1.5}$ edges outside expanders.

$O(1)$ expanders with $\Omega(n)$ nodes.

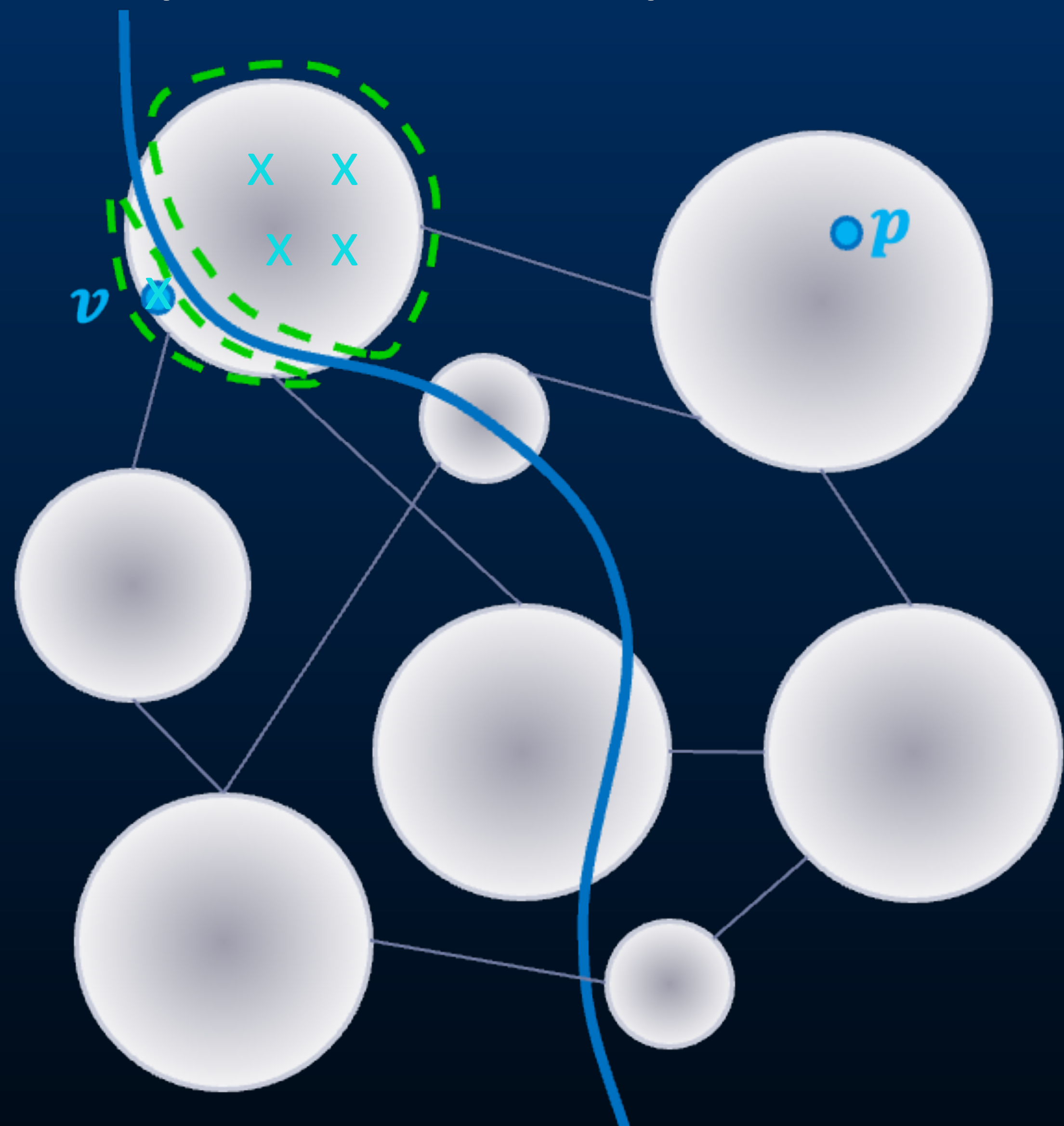
Simple graphs only:

$\leq \sqrt{n}$ nodes in small expanders.

Figure by Ohad Trabelsi.

Expanders-guided querying

An expander decomposition of G:



For each large expander:
Use Isolating Cuts with random terminals.

Repeat $\sqrt{n}$ *times.*

$\tilde{O}(n^{2.5})$ for simple graphs: [AKT STOC'21]

Set: $\phi = 1/\sqrt{n}$

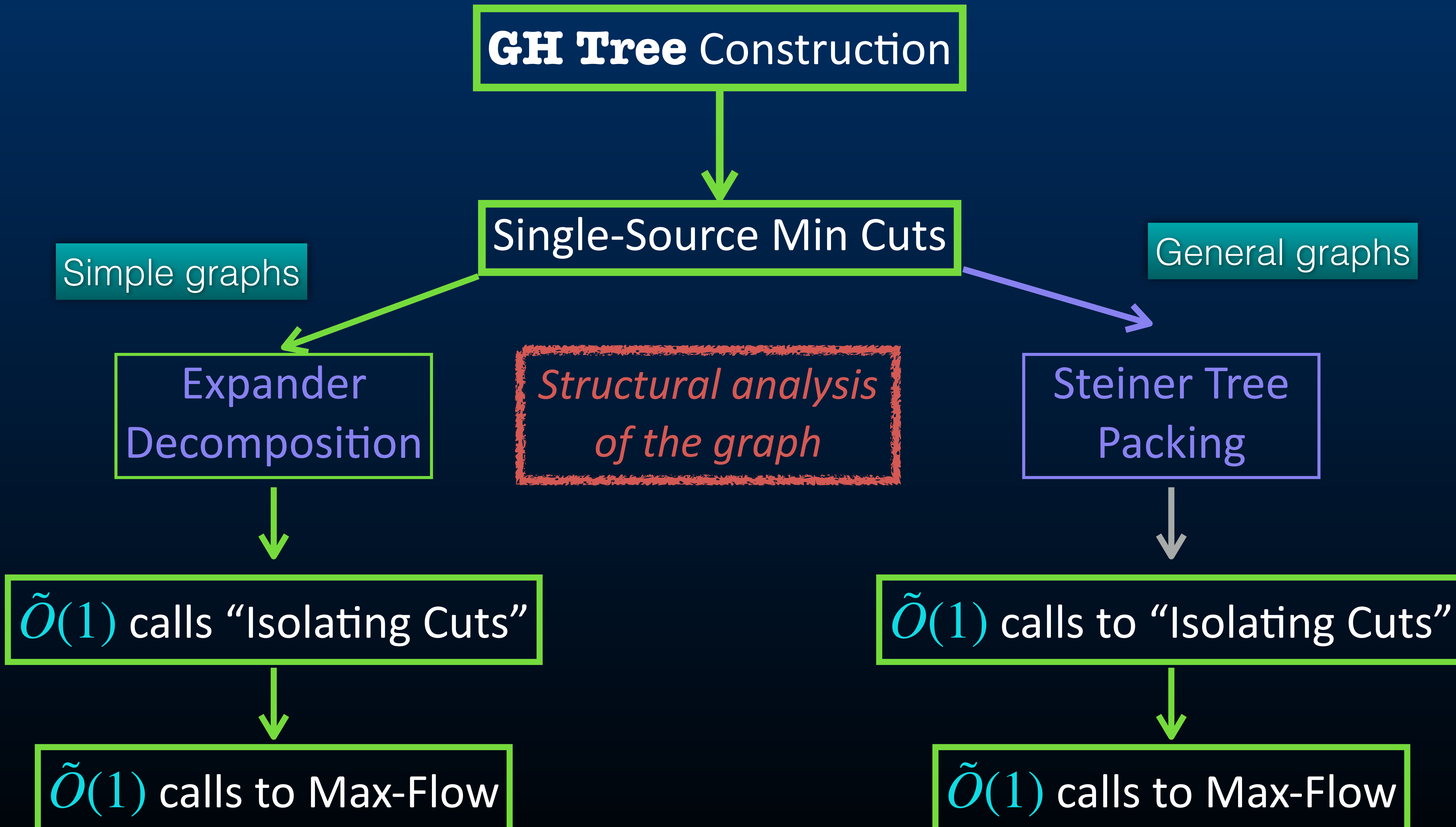
1. In any expander:
   $\leq \sqrt{n}$ nodes from one side
2. $\leq n^{1.5}$ edges outside expanders.

$O(1)$ expanders with $\Omega(n)$ nodes.

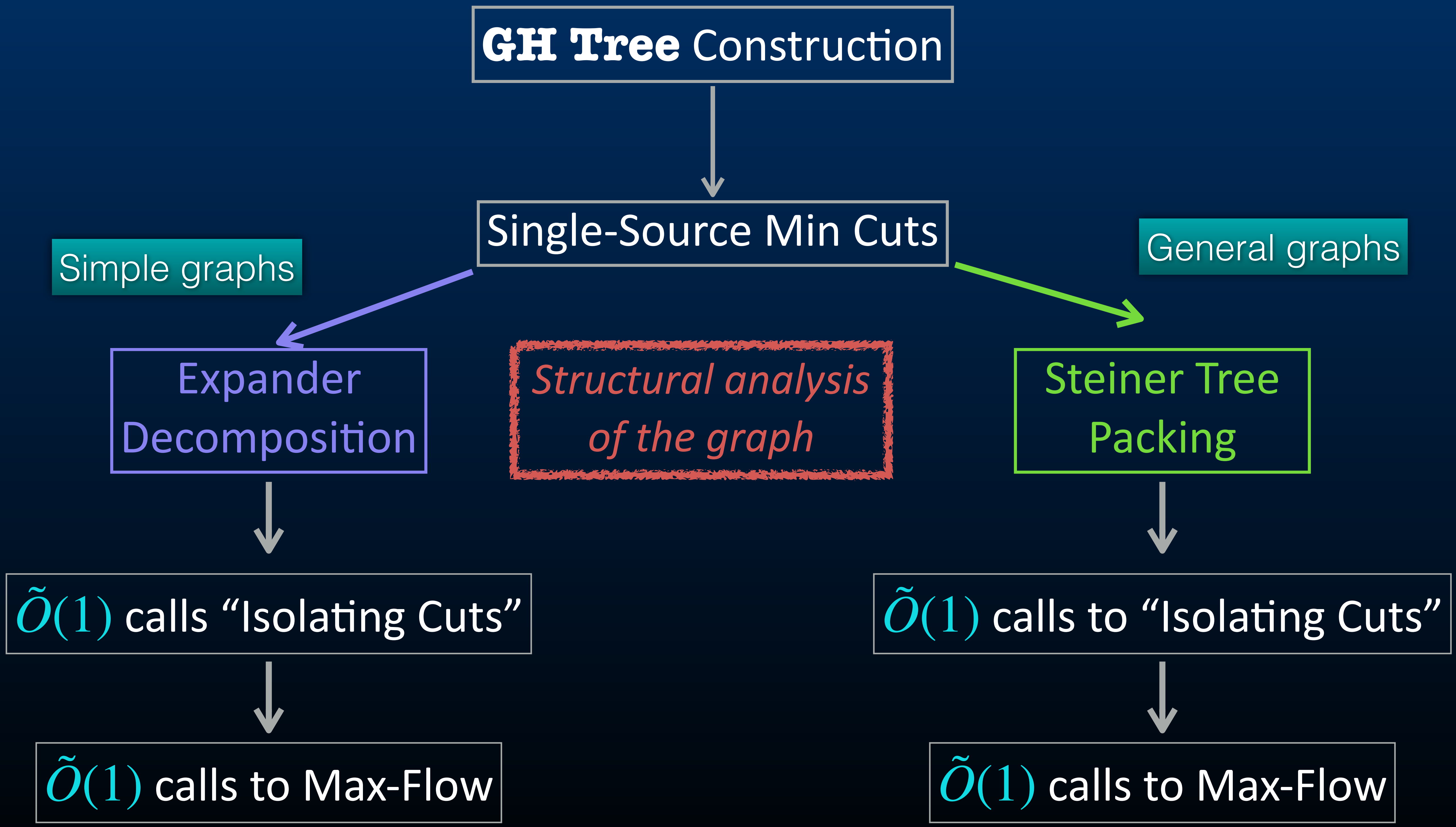
Simple graphs only:
$\leq \sqrt{n}$ nodes in small expanders.

Figure by Ohad Trabelsi.

Technical Overview



Technical Overview



Comparison with Global Min-Cut

Global Min Cut

Simple graphs

Expander Decomposition

[Kawarabayashi-Thorup'15]

General graphs

Spanning Tree Packing

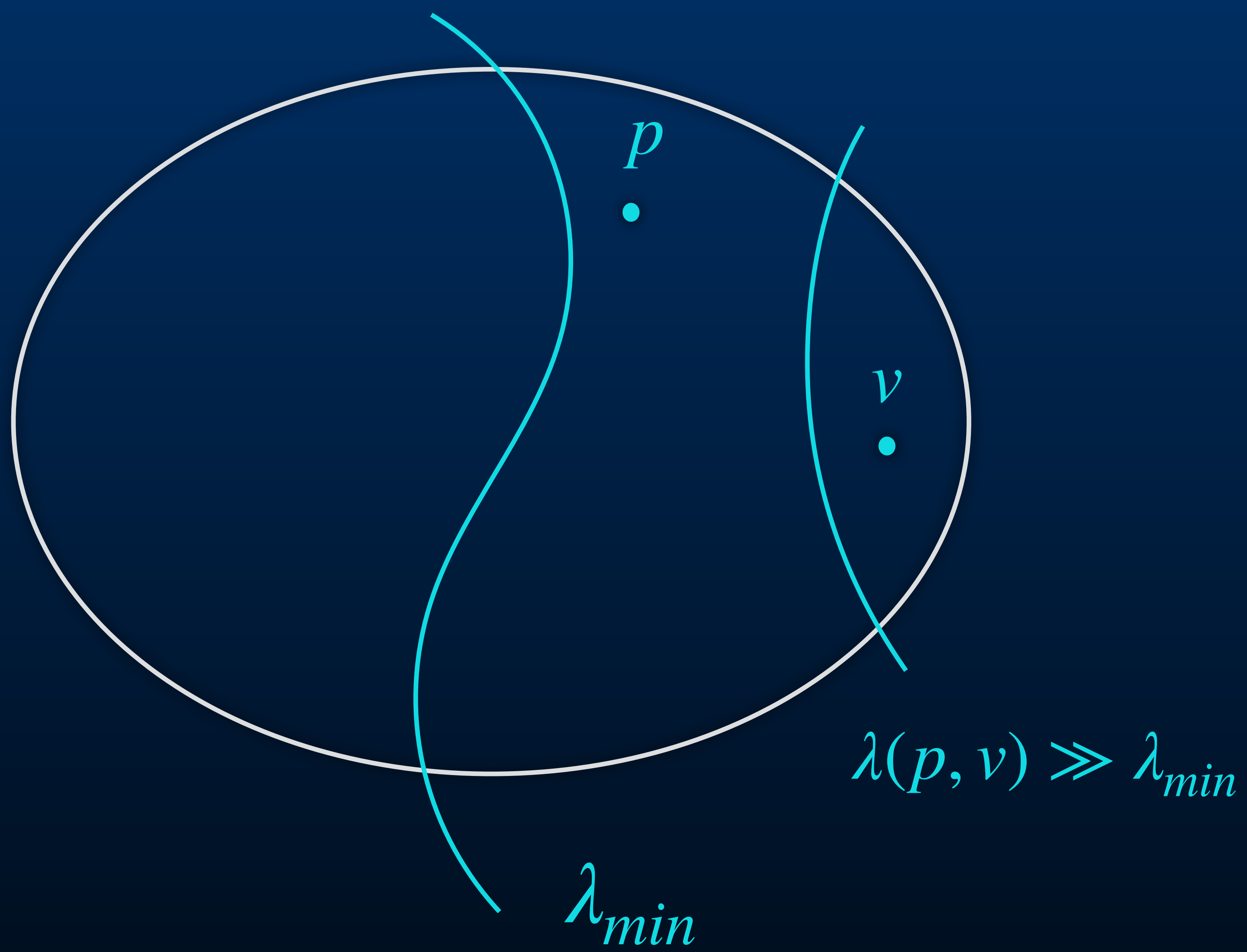
[Karger'96]

We generalize both techniques to single-source min-cuts…

<u>Main challenge:</u> existence of low degree nodes.

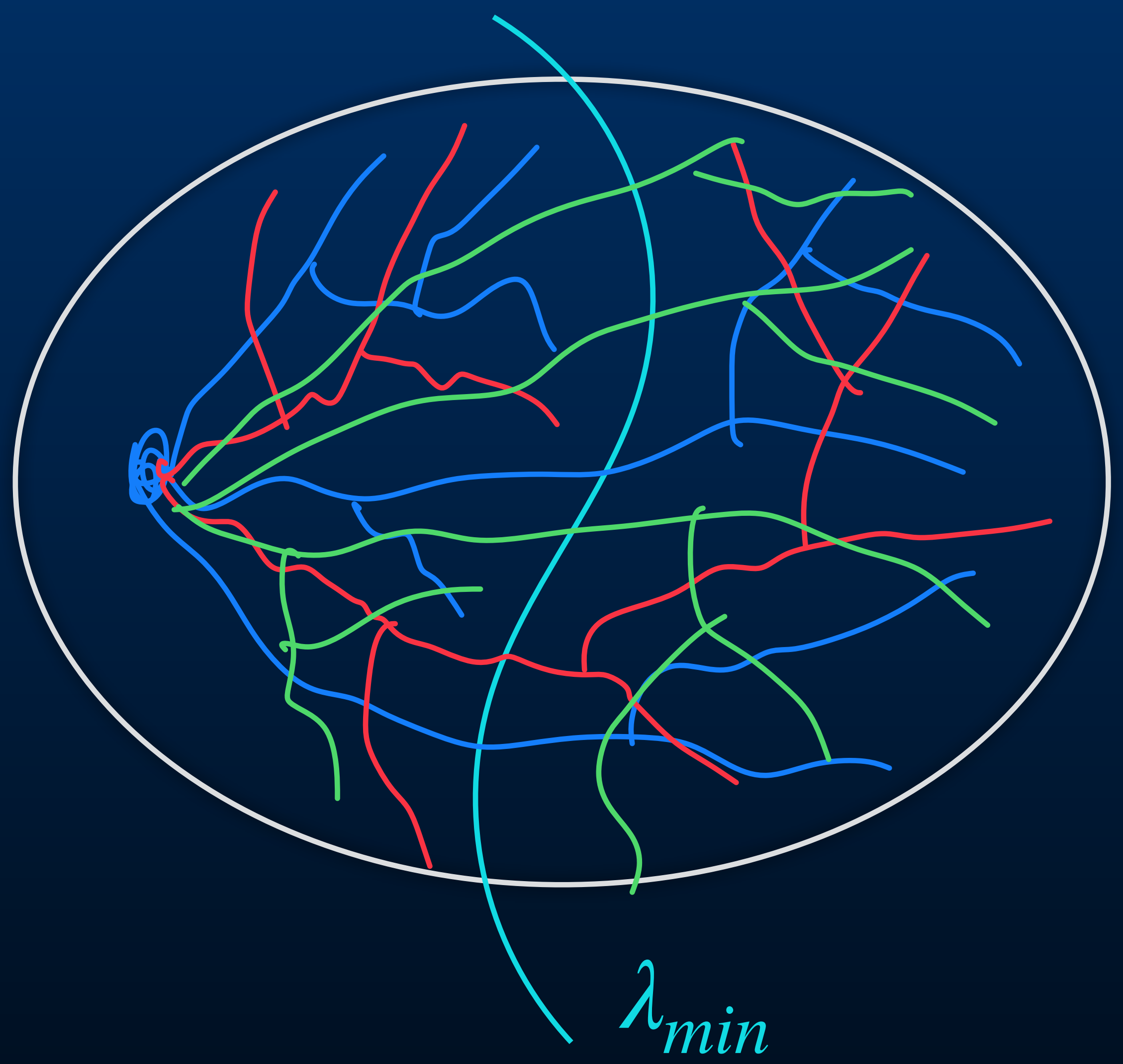
Spanning Tree Packing → ***Steiner*** Tree Packing

From Tree Packings to k-Respecting Trees



Main challenge: existence of low degree nodes.

From Tree Packings to k-Respecting Trees



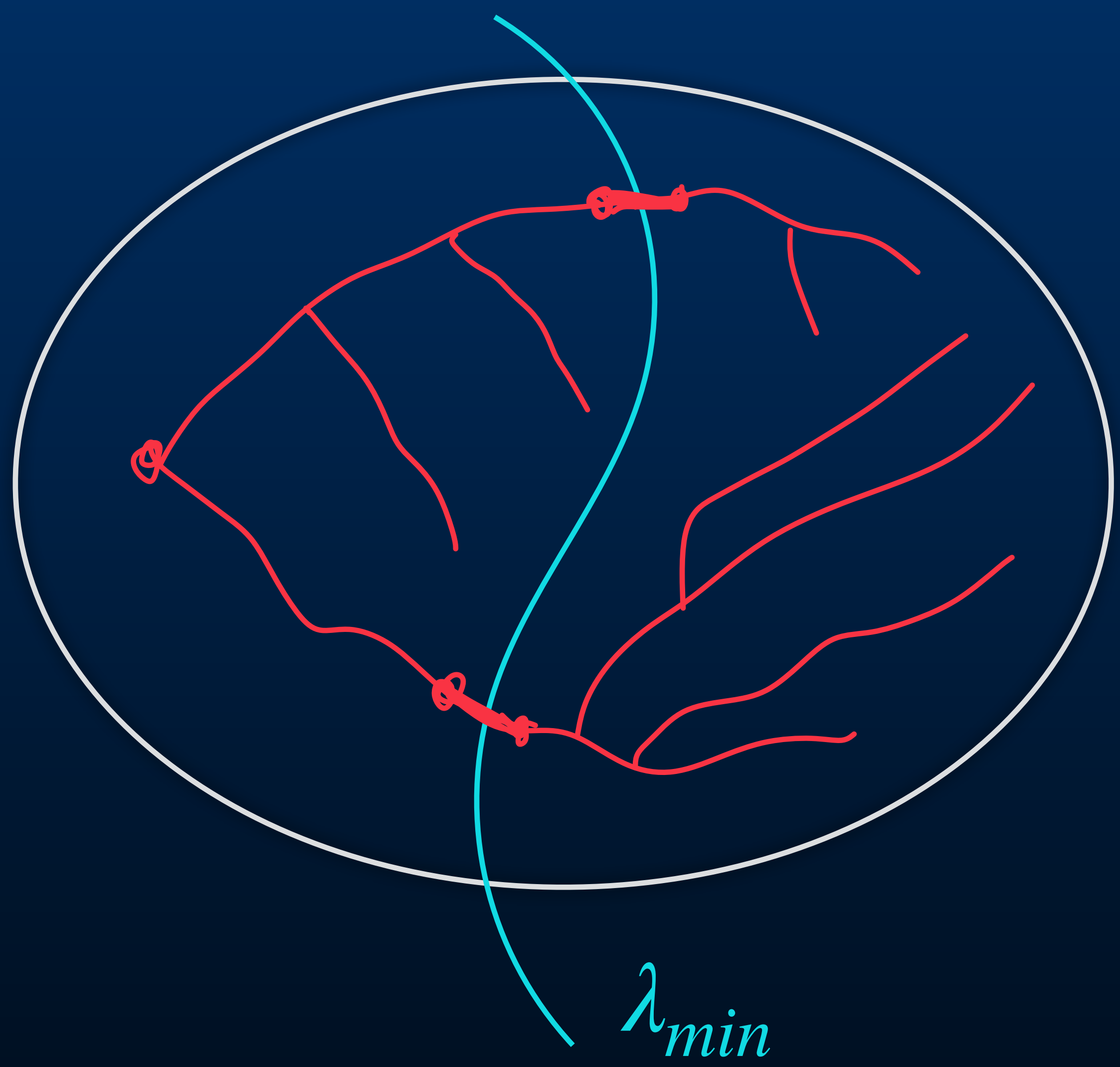
Thm [Nash-Williams and Tutte 1961]:

There are $\lambda_{min}/2$ disjoint spanning trees.

Karger's Algorithm:

1. Pack $\sim \lambda_{min}/2$ spanning trees.
2. Pick one at random, it is 2-respecting.
3. Solve min-cut with 2-respecting tree.

From Tree Packings to k-Respecting Trees



Thm [Nash-Williams and Tutte 1961]:

There are $\lambda_{min}/2$ disjoint spanning trees.

Karger's Algorithm:

1. Pack $\sim \lambda_{min}/2$ spanning trees.
2. Pick one at random, it is 2-respecting.
3. Solve min-cut with 2-respecting tree.

2-respecting spanning tree:

Up to 2 tree edges cut by min-cut.

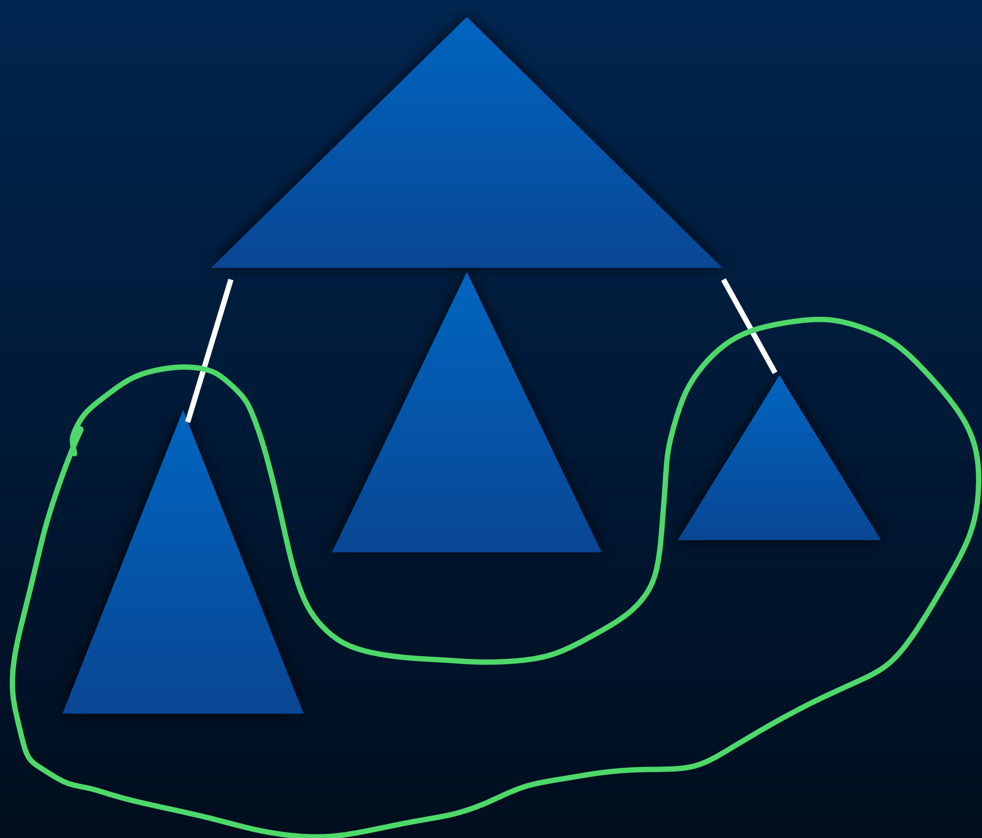
Low degree nodes prevent us from packing so many ***spanning*** *trees, but* ***Steiner*** *trees are OK.*

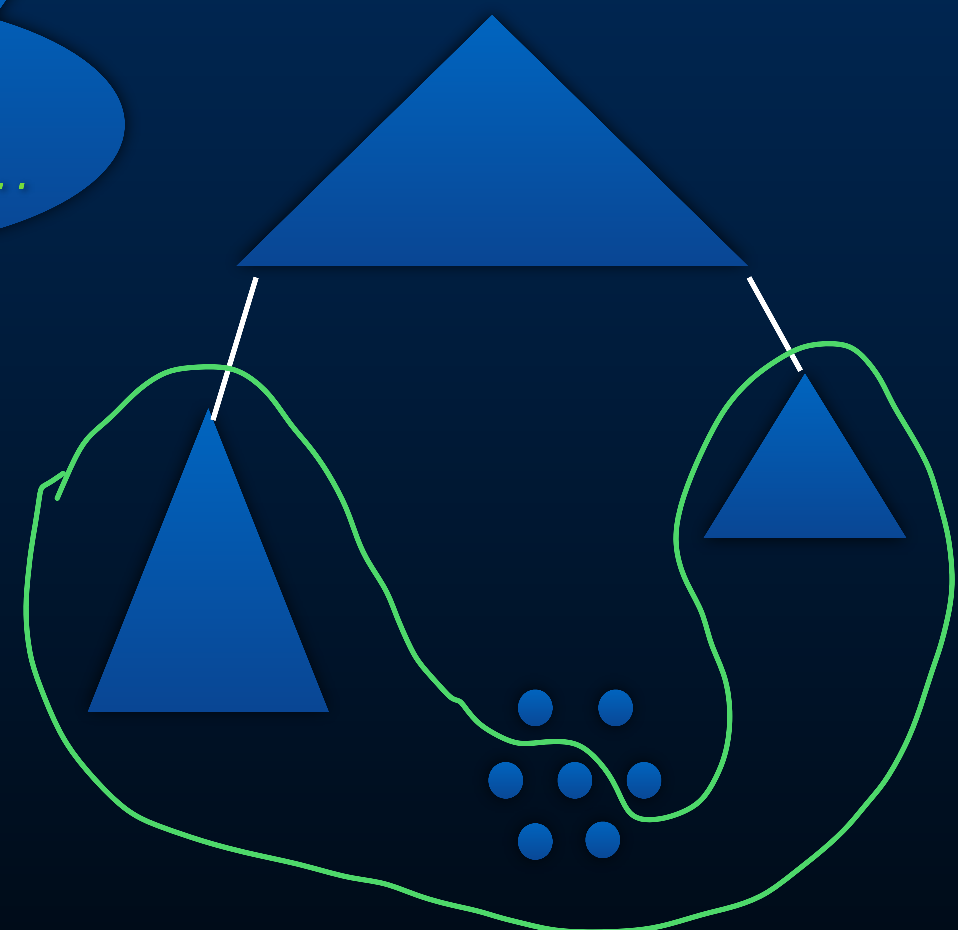
Spanning Trees vs. Steiner Trees

2-respecting spanning tree [Karger'96]:
≤ 2 tree edges cut by global min-cut.

2-respecting Steiner tree [AKLPST'21]:
≤ 2 tree edges cut by SS min-cuts.

Actually,
***4**-respecing…*

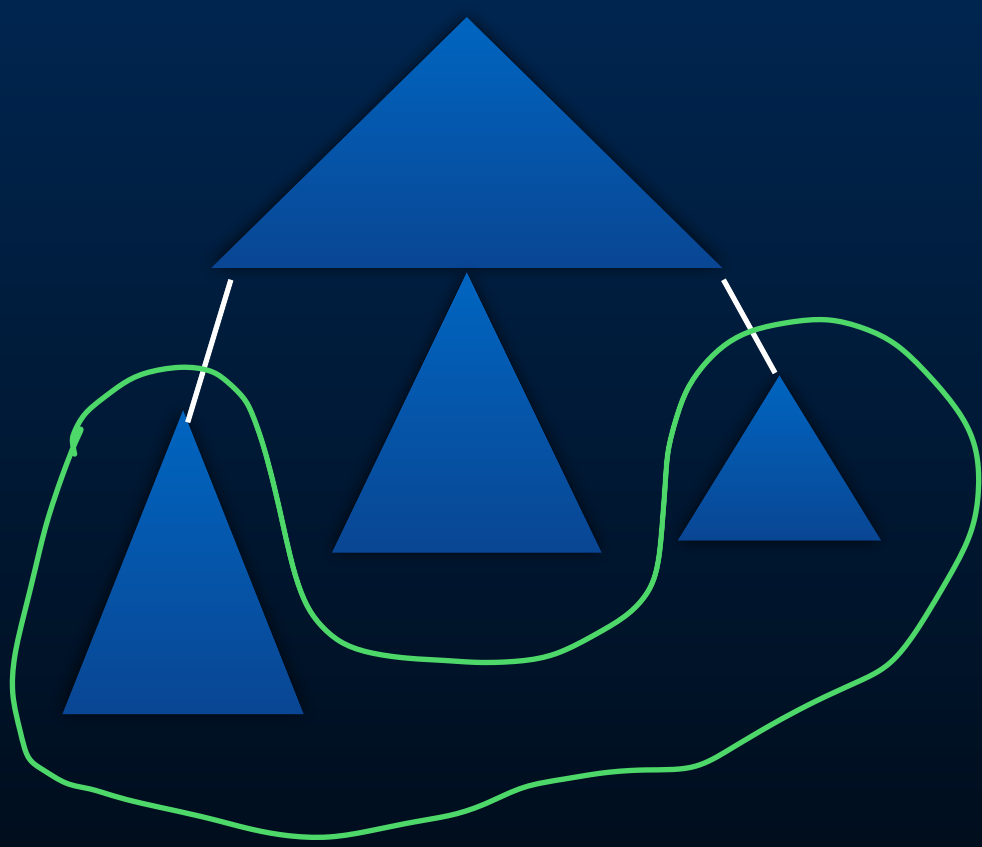




Spanning Trees vs. Steiner Trees

2-respecting <u>spanning</u> tree [Karger'96]:

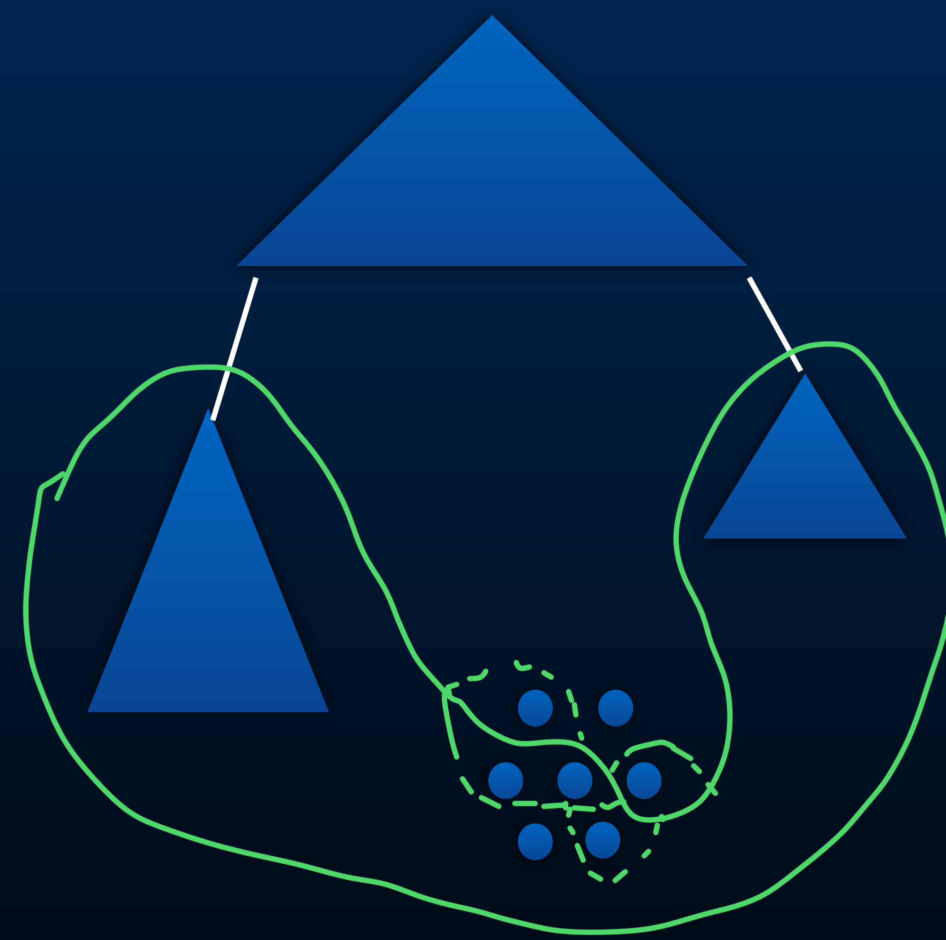
≤ 2 tree edges cut by global min-cut.



Cut fully determined by the 2 tree edges

2-respecting <u>Steiner</u> tree [AKLPST'21]:

≤ 2 tree edges cut by SS min-cuts.



Exponentially many options given the tree edges

SS Cuts via 4-respecting Steiner Trees

Our Algorithm:

There are $\lambda_{Steiner}/2$ disjoint Steiner trees.

Pack $\sim \lambda_{Steiner}/4$ Steiner trees, via MWU and Mehlhorn's 2-approximate Min-Steiner Tree Alg.

Sample a 4-respecting Steiner Tree.

Solve **SS-Min-Cuts** using 4-respecting Steiner tree, via **Isolating Cuts**.

Thm [Nash-Williams and Tutte 1961]:

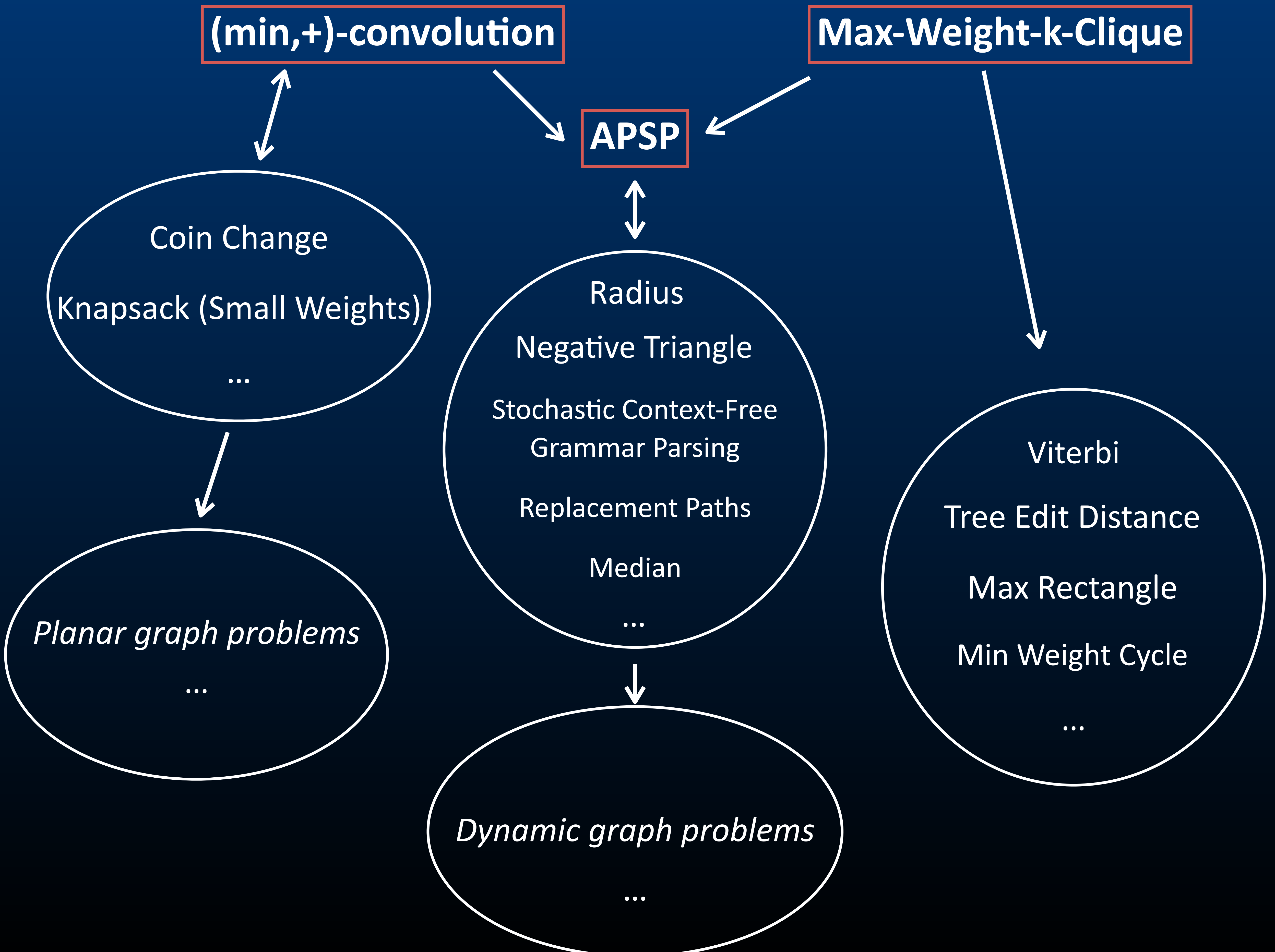
There are $\lambda_{min}/2$ disjoint spanning trees.

Karger's Algorithm:

1. Pack $\sim \lambda_{min}/2$ spanning trees.
2. Pick one at random, it is 2-respecting.
3. Solve min-cut with 2-respecting tree.

Open Questions

1. Refute the APSP Conjecture, or prove it. *e.g. under SETH/3SUM.*


(min,+)-convolution
Max-Weight-k-Clique
APSP
Coin Change
Knapsack (Small Weights)
...
Radius
Negative Triangle
Stochastic Context-Free Grammar Parsing
Replacement Paths
Median
...
Viterbi
Tree Edit Distance
Max Rectangle
Min Weight Cycle
...
Planar graph problems
...
Dynamic graph problems
...

Open Questions

1. Refute the APSP Conjecture, or prove it. *e.g. under SETH/3SUM.*

2. More applications of the techniques (Steiner Tree Packing, Isolating Cuts, Expander Decompositions with vertex demands)

   e.g. for k-cuts, vertex cuts, directed cuts, hypergraphs, etc.

3. **GH Tree** in $\tilde{O}$(Max-Flow-Time) for general graphs too?

4. Subcubic time deterministically for general graphs?

Thanks for your attention!