Breaking Quadratic Time for Small Vertex Connectivity and an Approximation Scheme

Danupon Nanongkai
KTH

Thatchaphol Saranurak
TTIC

Sorrachai Yingchareonthawornchai
Aalto University

@WOLA'19

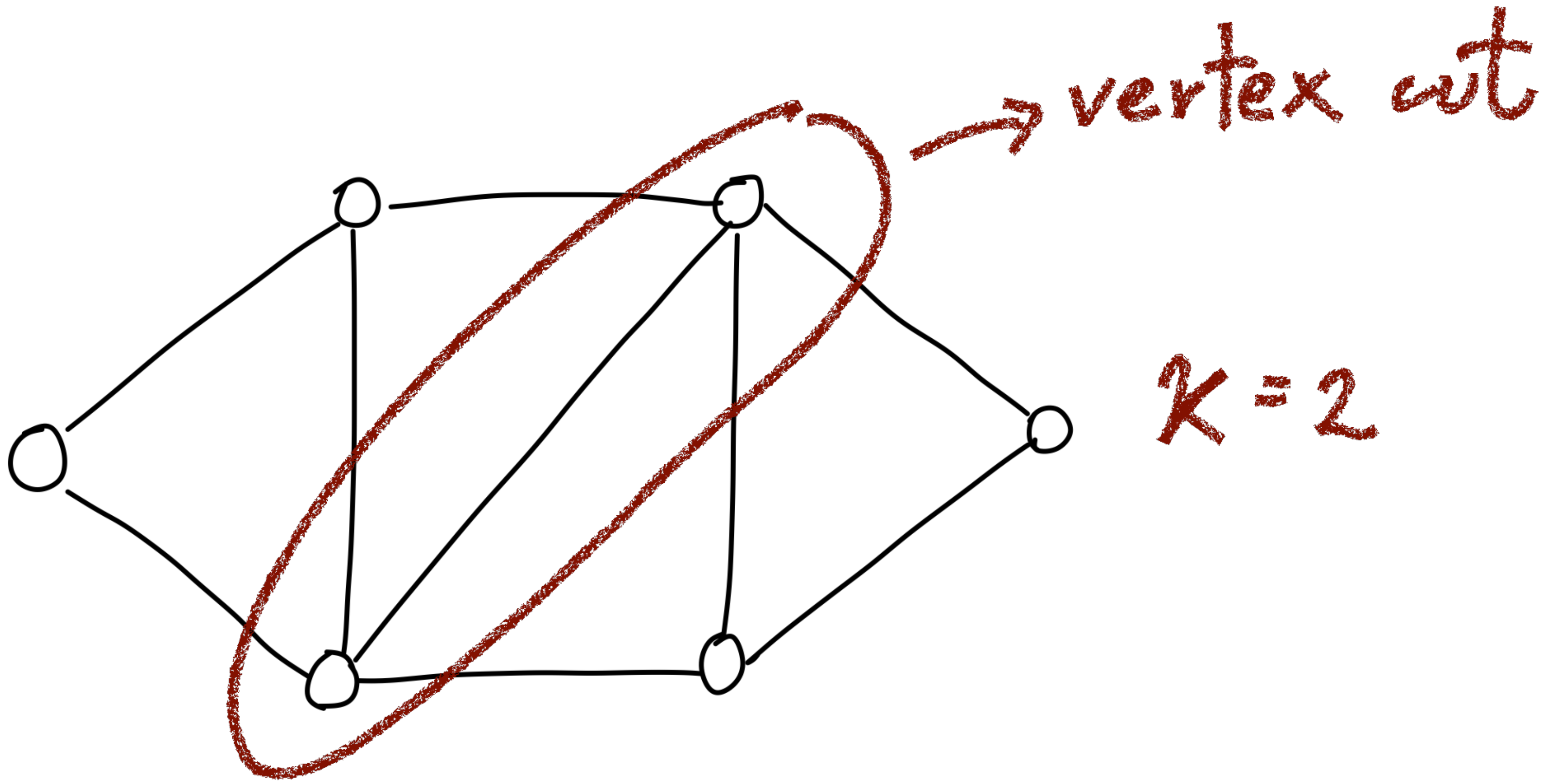
22 July 2019

Take Home Message:

“Local algorithms are useful for vertex connectivity”

Vertex Connectivity

$$G = (V, E), |V| = n, |E| = m$$



Compute κ the minimum number of vertices to be removed to disconnect the graph

Part 1: Introduction

Background and Results

$G = (V, E), |V| = n, |E| = m, k = \kappa$ *omit polylog factor

	Reference	General	$m = O(n)$
	Kleitman'69, Podderyugin'73, Even Tarjan'75	mnk^2	n^2
	Even'75, Galil'80, Esfahanian Hakimi'84, Matula'87	$(n + k^2)mk$	n^2
	Becker et al'82 (Randomized)	mnk	n^2
	Nagamochi Ibaraki'92	Can assume $m = O(nk)$	
*	Linial Lovasz Wigderson'88, Cheriyan Reif'91	$n^\omega + nk^\omega$	n^ω
*	Henzinger Rao Gabow'00 (Randomized)	mn	n^2

*** state-of-the-art**

$G = (V, E), |V| = n, |E| = m, k = \kappa$ *omit polylog factor

	Reference	General	$m = O(n)$
	Kleitman'69, Podderyugin'73, Even Tarjan'75	mnk^2	n^2
	Even'75, Galil'80, Esfahanian Hakimi'84, Matula'87	$(n + k^2)mk$	n^2
	Becker et al'82 (Randomized)	mnk	n^2
	Nagamochi Ibaraki'92	Can assume $m = O(nk)$	
*	Linial Lovasz Wigderson'88, Cheriyan Reif'91	$n^\omega + nk^\omega$	n^ω
*	Henzinger Rao Gabow'00 (Randomized)	mn	n^2
	Gabow'06 (Deterministic)	$mn + m\min\{k^{2.5} + n^{3/4}k\}$	n^2

Even for $m = O(n), k = O(1)$,
$O(n^2)$ time remains since 1969

$G = (V, E), |V| = n, |E| = m, k = \kappa$ *omit polylog factor

Reference	General	$m = O(n)$
Kleitman'69, Podderyugin'73, Even Tarjan'75	mnk^2	n^2
Even'75, Galil'80, Esfahanian Hakimi'84, Matula'87	$(n + k^2)mk$	n^2
Becker et al'82 (Randomized)	mnk	n^2
Nagamochi Ibaraki'92	Can assume $m = O(nk)$	
Linial Lovasz Wigderson'88, Cheriyan Reif'91	$n^\omega + nk^\omega$	n^ω
Henzinger Rao Gabow'00 (Randomized)	mn	n^2
Gabow'06 (Deterministic)	$mn + m \min\{k^{2.5} + n^{3/4}k\}$	n^2
Our work (STOC'19)	$m + n^{4/3}k^{7/3}$	$n^{4/3}$
Our follow-up work'19, Forster Young'19	$m + nk^3$	n

Near-linear time when k = O(polylog(n))

Approximation

$O(n^{2.5})$ **when** $k = \Omega(n^{0.64..})$

Reference	Time	Approx
Linial Lovasz Wigderson'88, Cheriyan Reif'91, Henzinger'97	$\min\{n^{2.5}, n^2k, n^\omega + nk^\omega\}$	2
Censor-Hillel Ghaffari Kuhn'14	m	$\log n$
Our work* (STOC'19)	$\min\{n^{2.2+o(1)}, n^\omega\}$	$(1+\epsilon)$

For any k

*Actual bound is $m + \frac{1}{\textbf{poly}(\epsilon)} \min\left\{nk^2, n^{5/3+o(1)}k^{2/3}, \frac{n^{3+o(1)}}{k}, n^\omega\right\}$

Remark: use fast approximate s,t-vertex connectivity from Chuzhoy Khanna (STOC'19)

Contributions

(including our follow-up work)

Breaking 50-year-old $O(n^2)$ time for small κ

Near-linear time for $\kappa = O(1)$

Fastest up to $\kappa = O(n^{0.457})$

Fastest $(1+\epsilon)$-approximation for any κ

Follow-up'19:
Testing *k*-Vertex Connectivity

One-sided tester: *k*-vertex connected vs. ϵ-far

Reference	Query Complexity (Simplified)	
Goldreich Ron'02, Orenstein Ron'11, Yoshida Ito'10, Yoshida' Ito 12	$\tilde{O}(\frac{2^{k+1}}{\epsilon^{k+1}})$	**Open**: polynomial #queries?
Forster Young'19	$\tilde{O}(\frac{k^3}{\epsilon^2})$	
Our follow-up '19	$\tilde{O}(\frac{k}{\epsilon^2})$	

Part 2: Techniques

$$G = (V, E), |V| = n, |E| = m, k = \kappa$$ *omit polylog factor

Reference	General	$m = O(n)$
Kleitman'69, Podderyugin'73, Even Tarjan'75	mnk^2	n^2
Even'75, Galil'80, Esfahanian Hakimi'84, Matula'87	$(n + k^2)mk$	n^2
Becker et al'82	mnk	n^2
Nagamochi Ibaraki'92	Can assume $m = O(nk)$	
Linial Lovasz Wigderson'88, Cheriyan Reif'91	$n^\omega + nk^\omega$	n^ω
Henzinger Rao Gabow'00 (Randomized)	mn	n^2
Gabow'06 (Deterministic)	$mn + mk^{2.5} + mn^{3/4}k$	n^2

Barrier: Single-Source Vertex Connectivity

No $o(n^2)$ algorithm is known

Fix a node *s*, compute the smallest vertex-cut separating *s* from some node

Part 2.1:
Local Vertex Connectivity

to replace Single-Source Vertex Connectivity

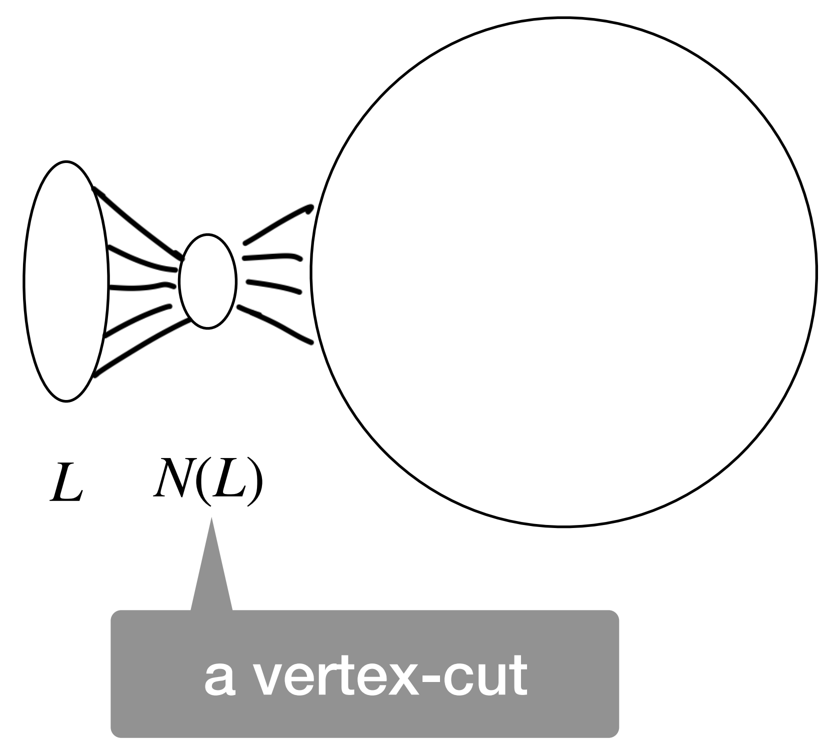
Fix a node *s*, is there a vertex-cut "near" *s* ?
(in sub-linear time)

Explore around s

Assumption: G **is regular,**

$$m = O(n), \kappa = O(1)$$

Notation: $N(L) =$ **neighbors of** $L \subset V$



Local Vertex Connectivity

Explore around x

Input: LocalVC$(x \in V, \nu, k)$ **where** $\nu < \frac{m}{k}$

Target vertex-cut

Output: Either

(1) declare that no $L \subset V$ **exists where** $L \ni x, |L| \leq \nu, |N(L)| < k,$

OR

(2) a vertex-set L **where** $|N(L)| < k$

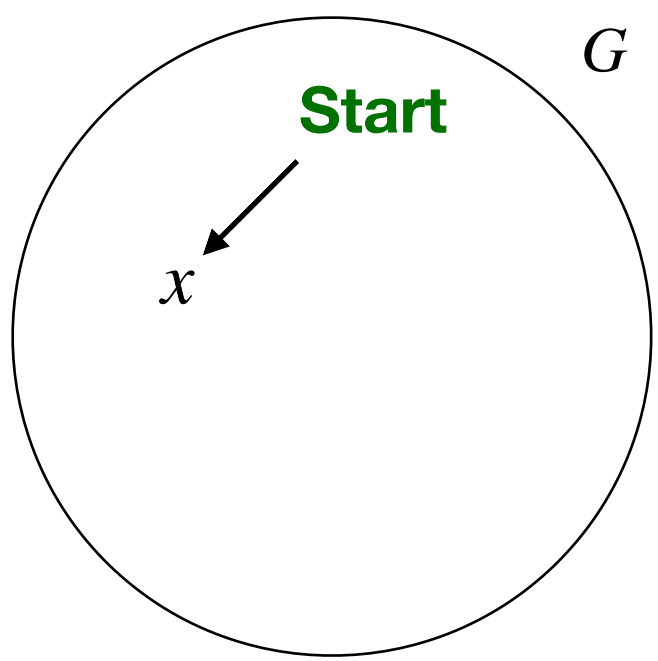
Local Vertex Connectivity

Explore around x

Input: LocalVC$(x \in V, \nu, k)$ **where** $\nu < \frac{m}{k}$

Output: Either

(1) declare that no $L \subset V$ **exists where** $L \ni x, |L| \le \nu, |N(L)| < k,$



OR

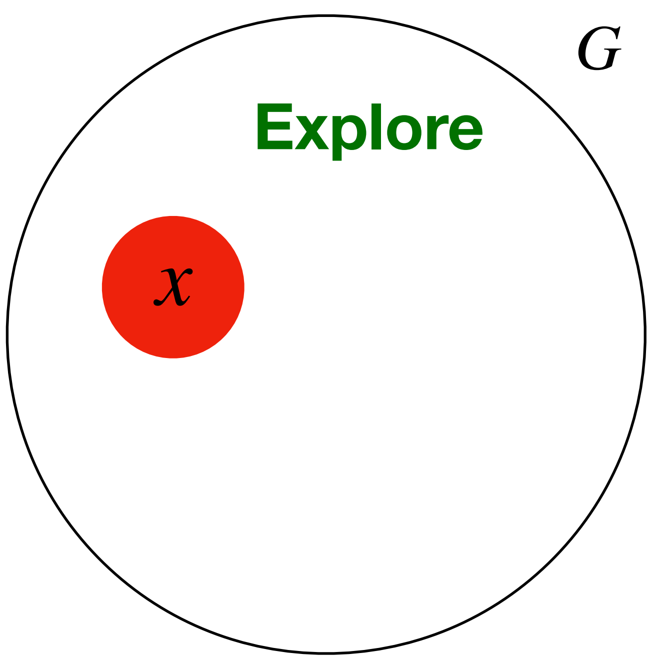
(2) a vertex-set L **where** $|N(L)| < k$

Local Vertex Connectivity

Input: LocalVC$(x \in V, \nu, k)$ **where** $\nu < \frac{m}{k}$

Output: Either

(1) declare that no $L \subset V$ **exists where** $L \ni x, |L| \leq \nu, |N(L)| < k,$



OR

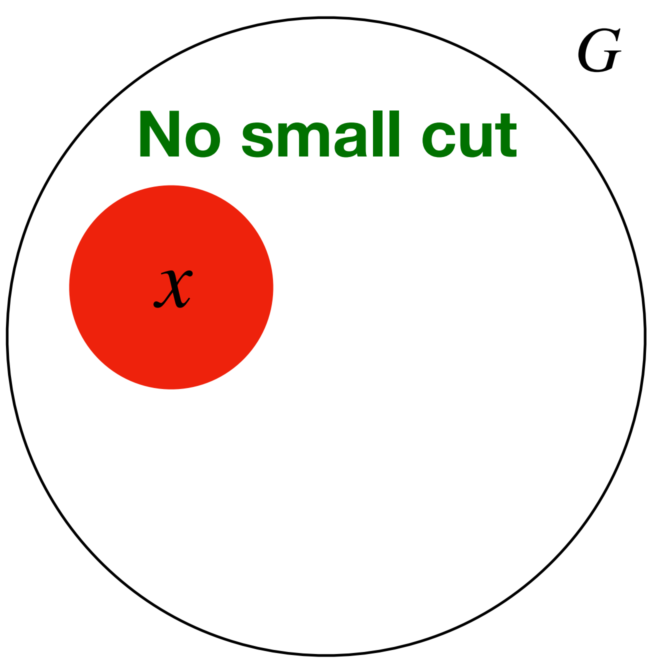
(2) a vertex-set L **where** $|N(L)| < k$

Local Vertex Connectivity

Input: LocalVC$(x \in V, \nu, k)$ **where** $\nu < \frac{m}{k}$

Output: Either

(1) declare that no $L \subset V$ **exists where** $L \ni x, |L| \leq \nu, |N(L)| < k,$



OR

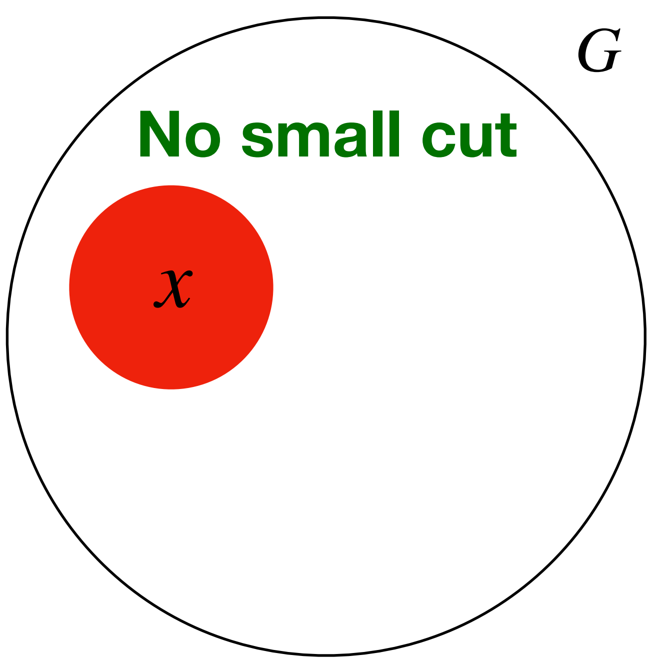
(2) a vertex-set L **where** $|N(L)| < k$

Local Vertex Connectivity

Input: LocalVC$(x \in V, \nu, k)$ **where** $\nu < \frac{m}{k}$

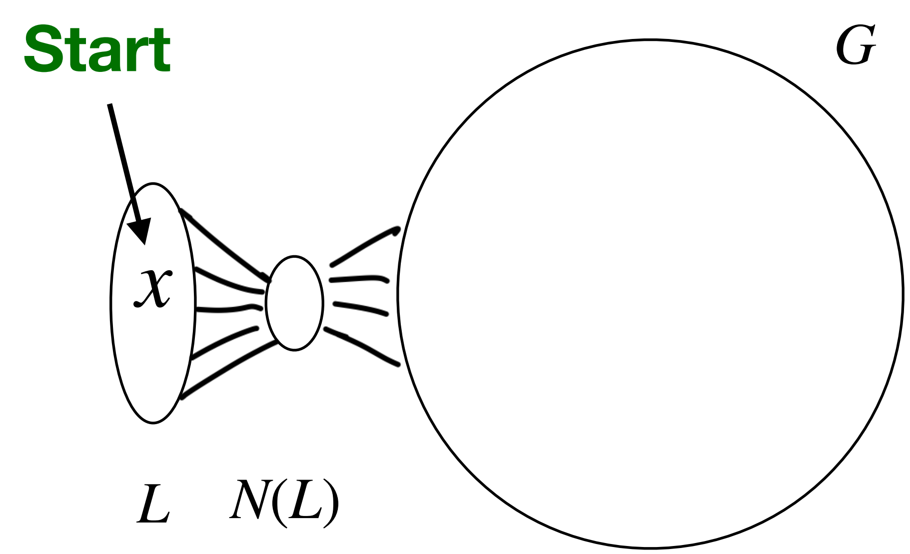
Output: Either

(1) declare that no $L \subset V$ **exists where** $L \ni x, |L| \le \nu, |N(L)| < k,$



OR

(2) a vertex-set L **where** $|N(L)| < k$

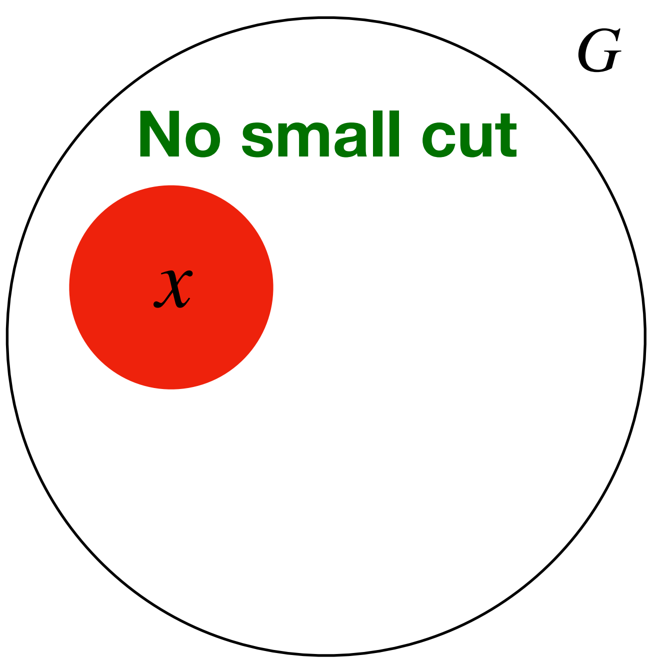


Local Vertex Connectivity

Input: LocalVC$(x \in V, \nu, k)$ **where** $\nu < \dfrac{m}{k}$

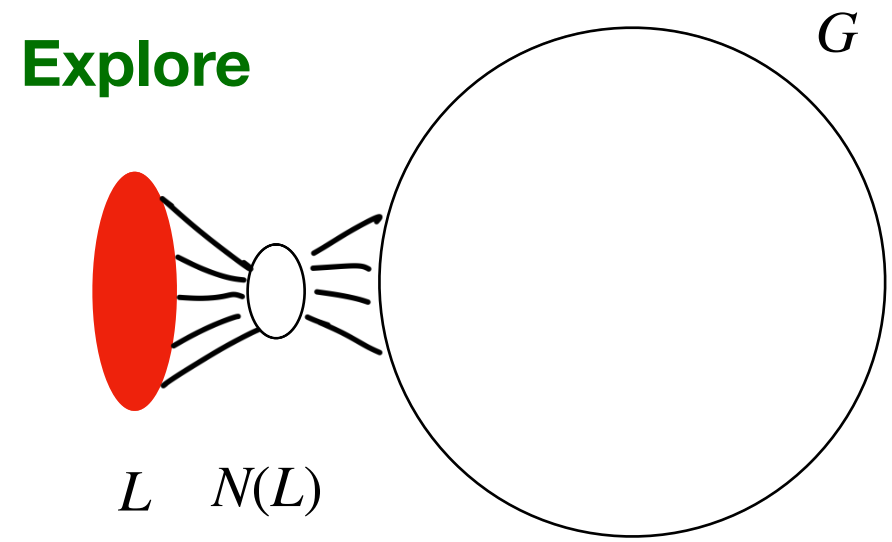
Output: Either

(1) declare that no $L \subset V$ **exists where** $L \ni x, |L| \leq \nu, |N(L)| < k,$



OR

(2) a vertex-set L **where** $|N(L)| < k$

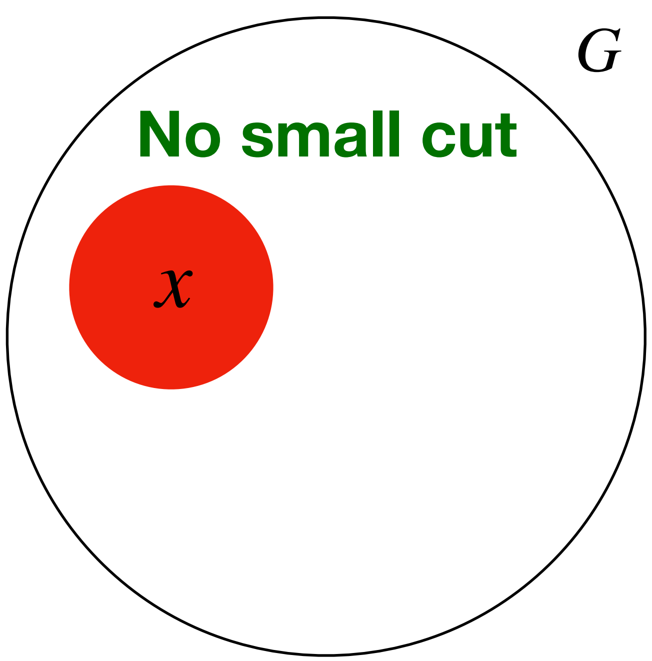


Local Vertex Connectivity

Input: LocalVC$(x \in V, \nu, k)$ **where** $\nu < \dfrac{m}{k}$

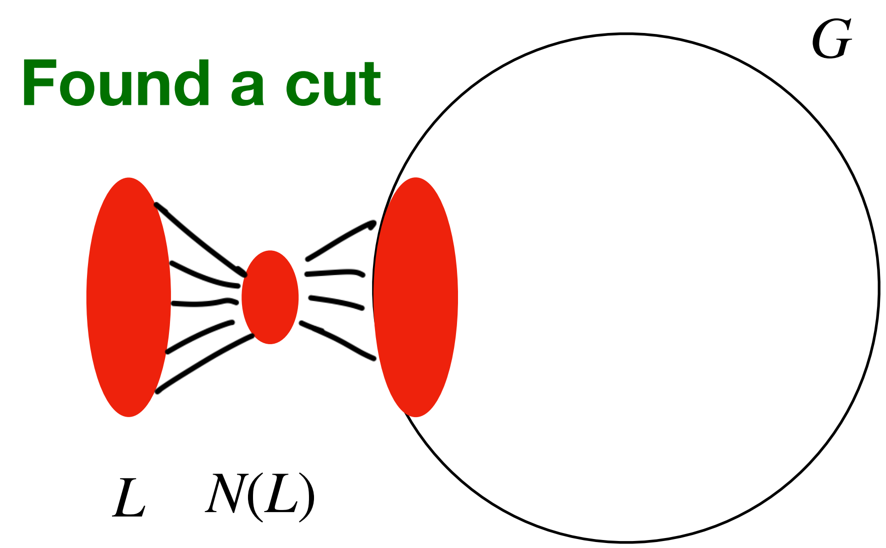
Output: Either

(1) declare that no $L \subset V$ **exists where** $L \ni x, |L| \leq \nu, |N(L)| < k,$



OR

(2) a vertex-set L **where** $|N(L)| < k$



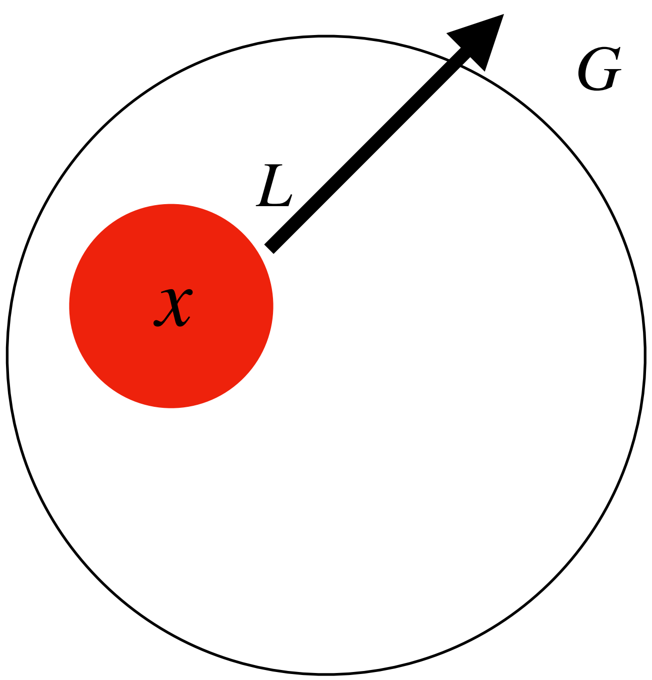
Local V… …nnectivity

Input: LocalVC(…

Output: Either

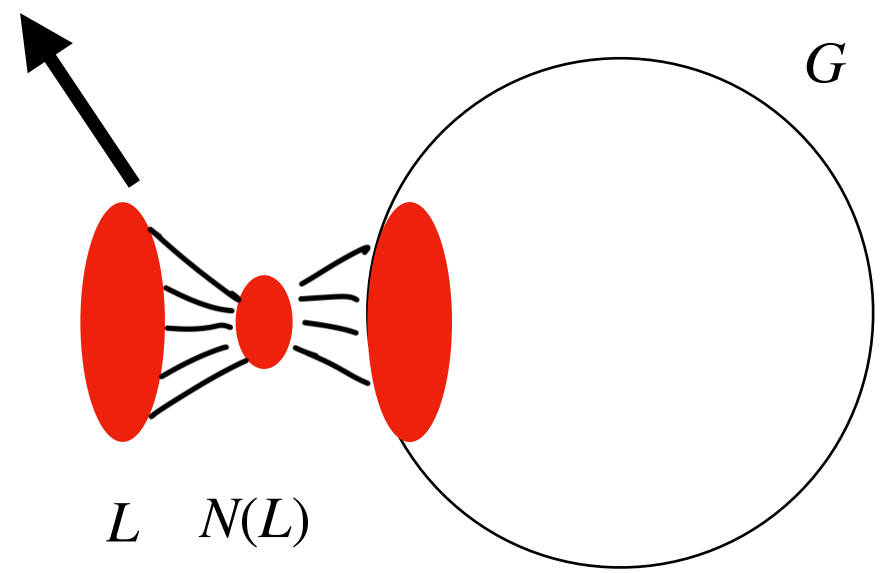
(1) declare that n…

where $L \ni x, |L| \le$ …



…2) a vertex-set L

where $|N(L)| < k$



Reading

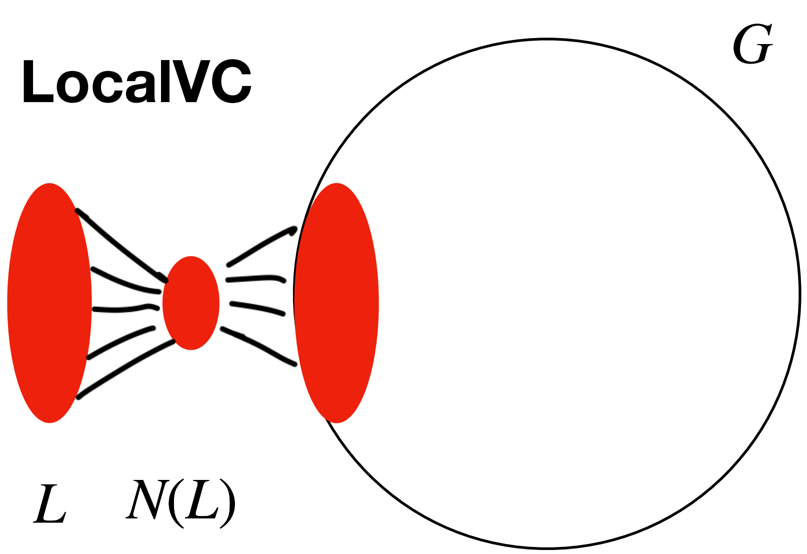
$O(\nu k)$

edges

possible

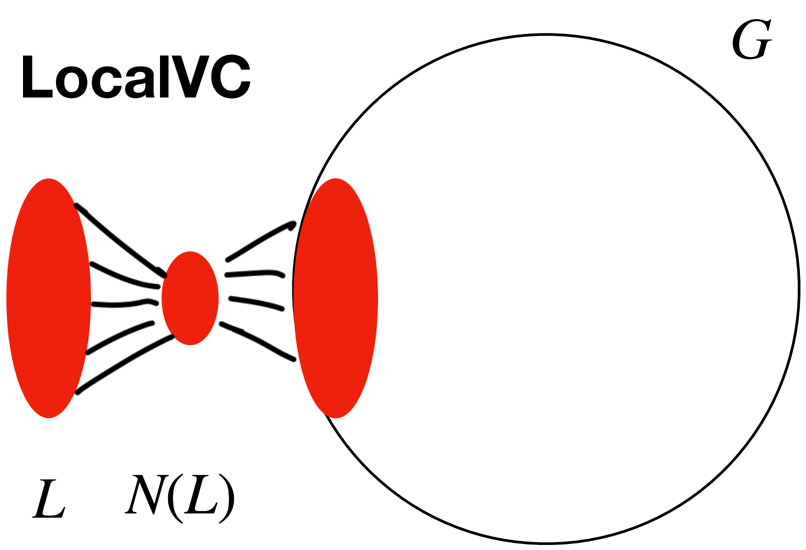
History

LocalVC



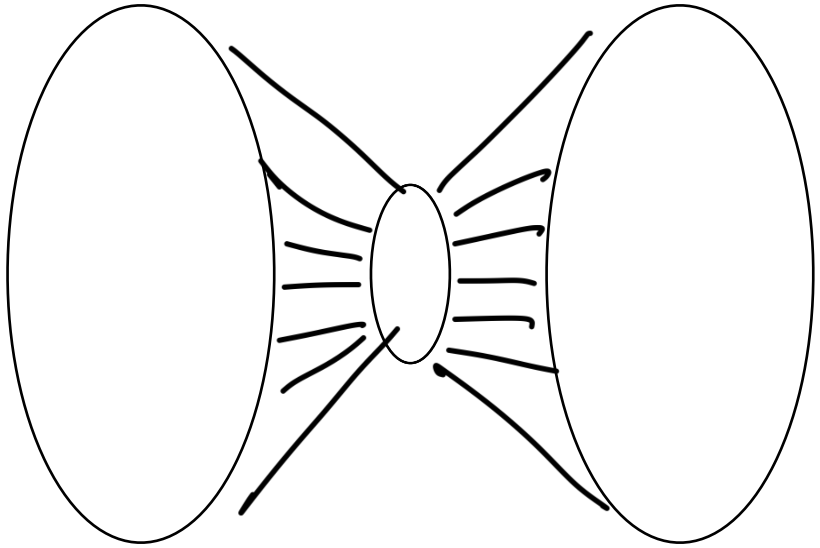
	LocalVC	Approximate LocalVC	Deterministic
This work (STOC '19)	$\nu^{1.5}k$	$\frac{\nu^{1.5}}{\epsilon^{1.5}\sqrt{k}}$	Yes
Adaptation of (Chechik, Hansen, Italiano, Loitzenbauer, Parotsidis SODA'17)	νk^k	-	Yes
Forster Young'19	νk^2	-	No
Our follow-up'19	νk^2	$\frac{\nu k}{\epsilon}$	No

Part 2.2:
Vertex Connectivity
via LocalVC



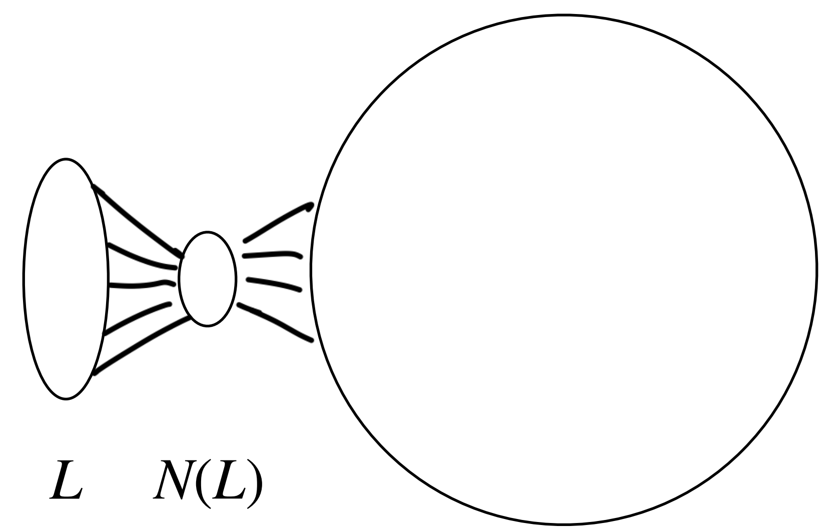
Suppose *G* has a vertex cut of size less than *k*

Highly balanced



$|L| \geq n/10$ $|R| \geq n/10$

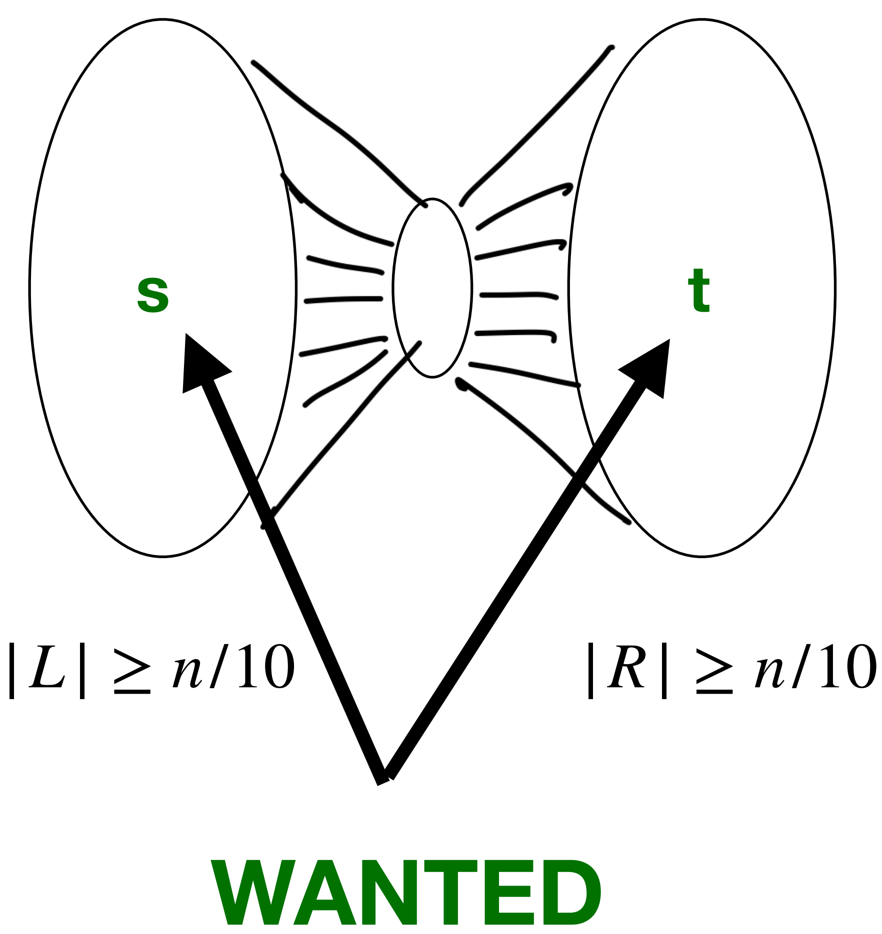
Highly lopsided



$|L| = \sqrt{n}$

Suppose *G* has a vertex cut of size less than *k*

Highly balanced



Fact: Compute s, t-connectivity by s, t-Max-Flow in $O(mk) = O(n)$

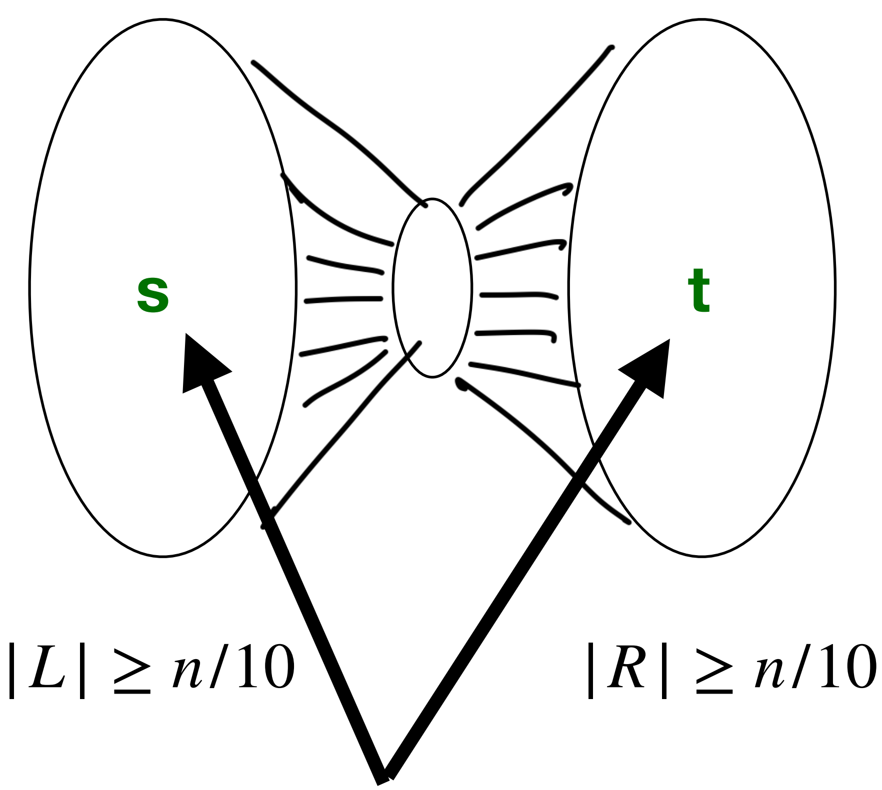
(e.g. Ford Fulkerson)

Suppose $|L| \geq n/10, |R| \geq n/10$

Compute s, t-connectivity

Suppose G has a vertex cut of size less than k

Highly balanced



WANTED

Fact: Compute s, t-connectivity by s, t-Max-Flow in $O(mk) = O(n)$

Suppose $|L| \geq n/10, |R| \geq n/10$

Sample $O(1)$ **pairs of nodes**

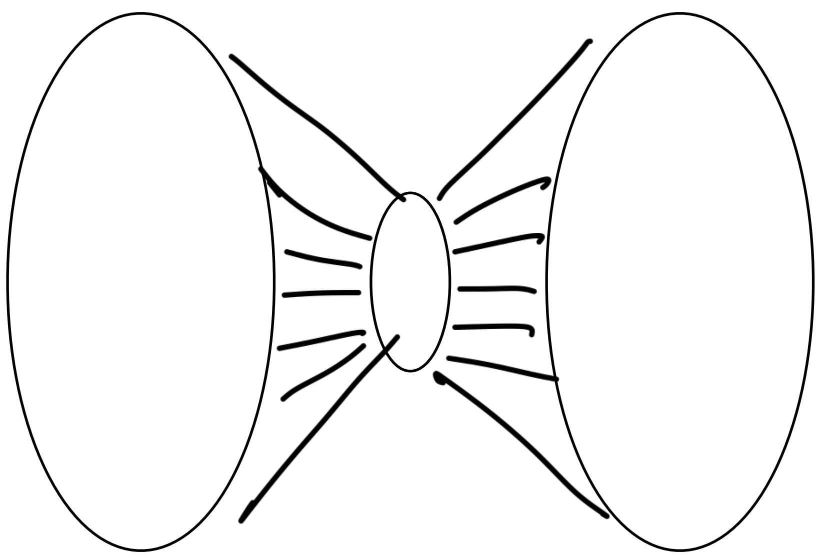
For each pair,

Compute s, t-connectivity

Total Time $= O(1) \times O(n) = O(n)$

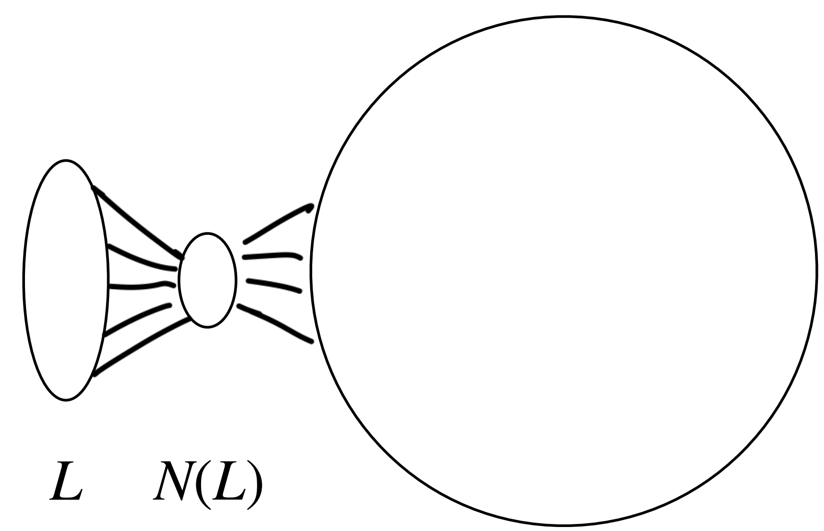
Suppose G has a vertex cut of size less than k

Highly balanced



$|L| \geq n/10$ $|R| \geq n/10$

Highly lopsided



$|L| = \sqrt{n}$

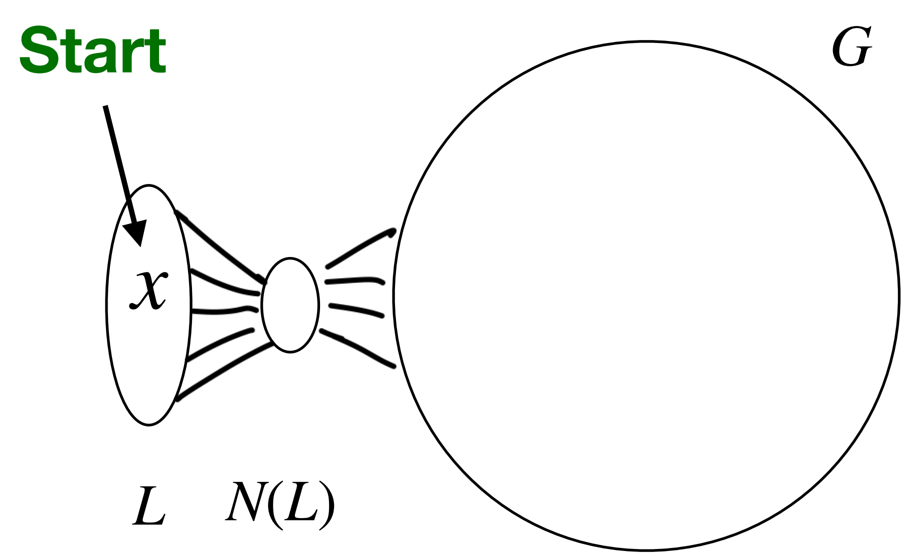
Suppose G has a vertex cut of size less than k

New

Theorem: Local Vertex Connectivity

$|L| \leq \nu, |N(L)| < k$ **in** $O(\nu k^2)$

Highly lopsided



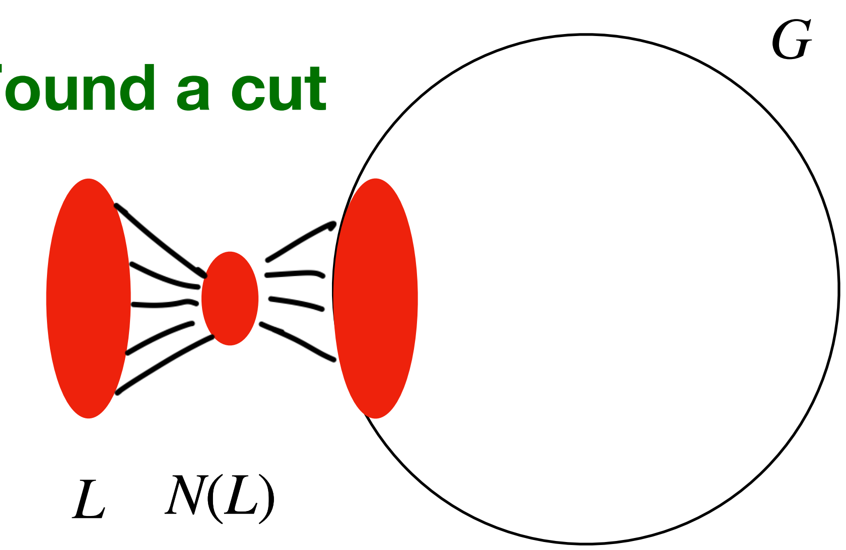
Suppose $|L| = \sqrt{n}$

Compute LocalVC$(x, \nu := \sqrt{n}, k)$

Suppose G has a vertex cut of size less than k

Highly lopsided

Found a cut



New

> **Theorem: Local Vertex Connectivity**
>
> $|L| \leq \nu, |N(L)| < k$ **in** $O(\nu k^2)$

Suppose $|L| = \sqrt{n}$

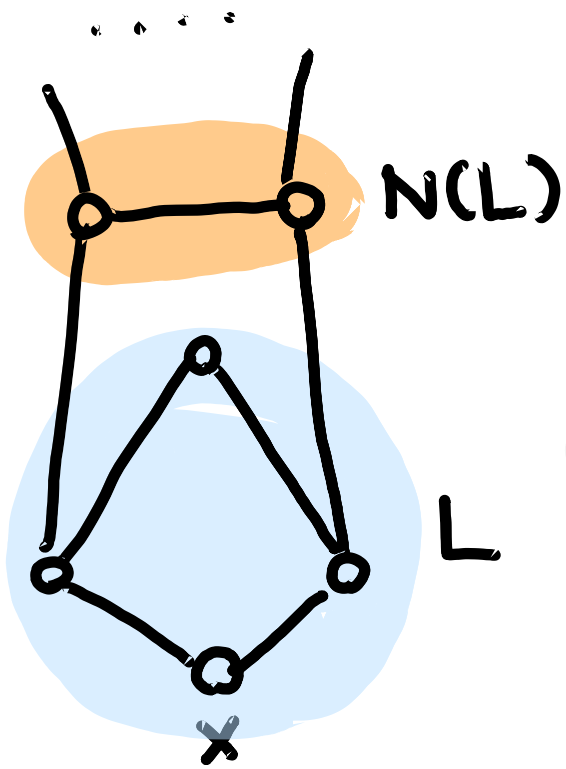
Sample $O(\sqrt{n})$ **nodes**

For each node,

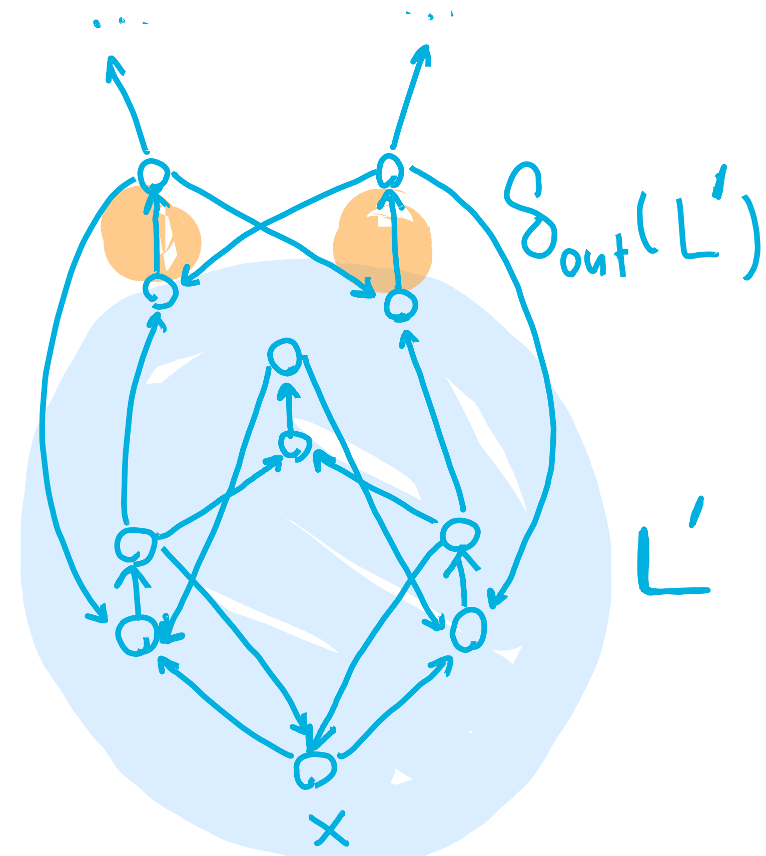
Compute LocalVC$(x, \nu := \sqrt{n}, k)$

Total Time $= O(\sqrt{n}) \times O(\sqrt{n}k^2) = O(n)$

Part 3: Local Vertex Connectivity in $O(\nu k^2)$



To

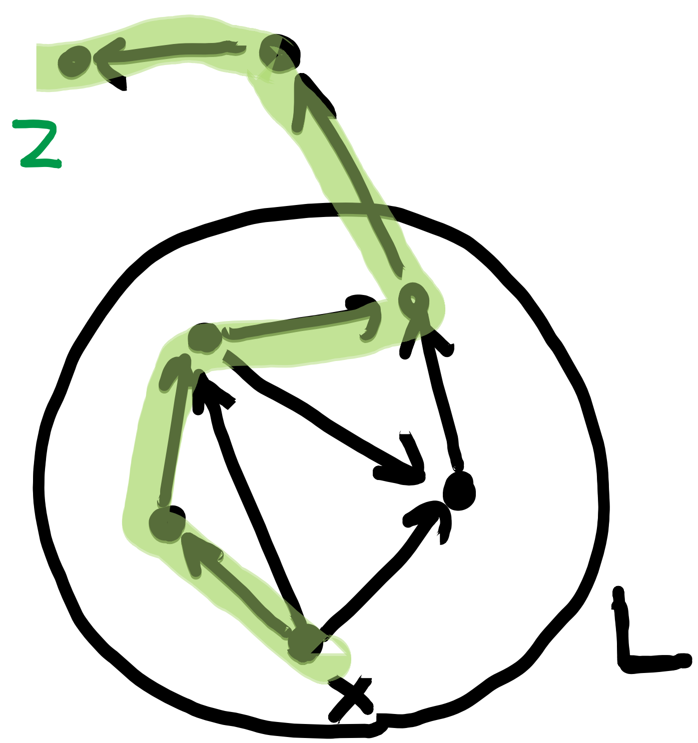


Vertex Connectivity

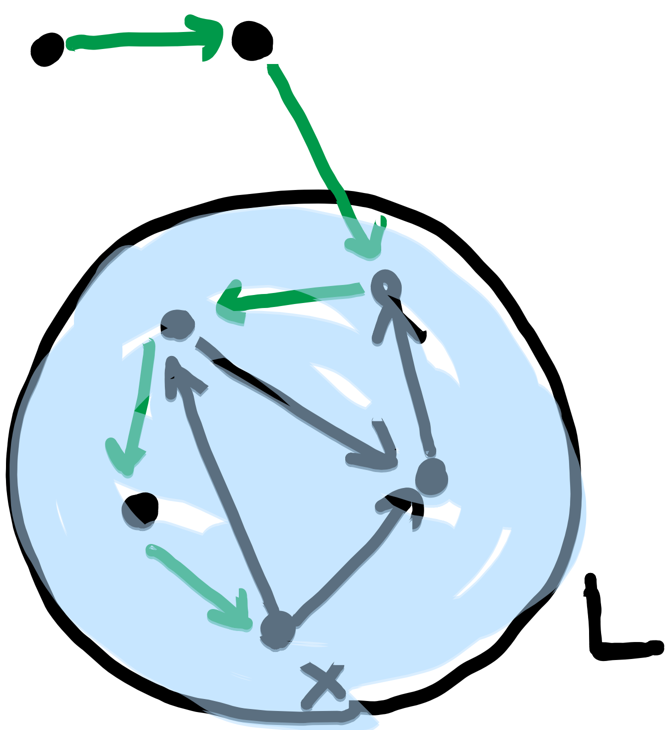
Directed Edge Connectivity

Local Edge Connectivity

Suppose *L* has 1 leaving edge



DFS exactly
$\nu + 1$
edges



1. Reverse x,z-path
2. DFS will get stuck
and we get *L*

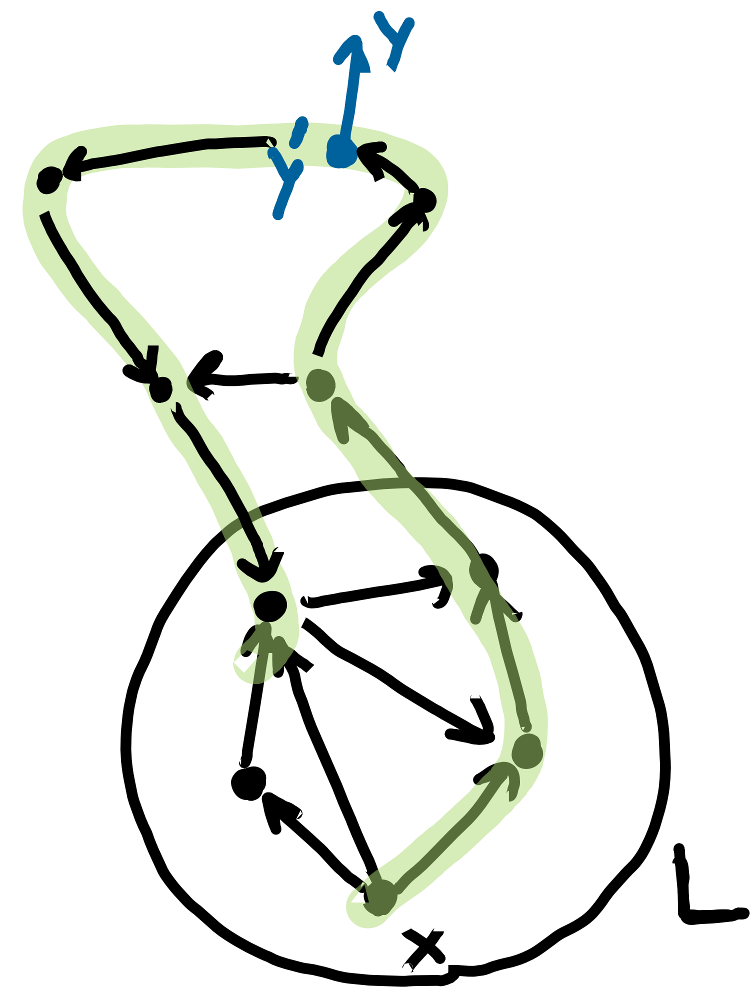
Local Edge Connectivity

Complete pseudocode:

Repeat *k* times

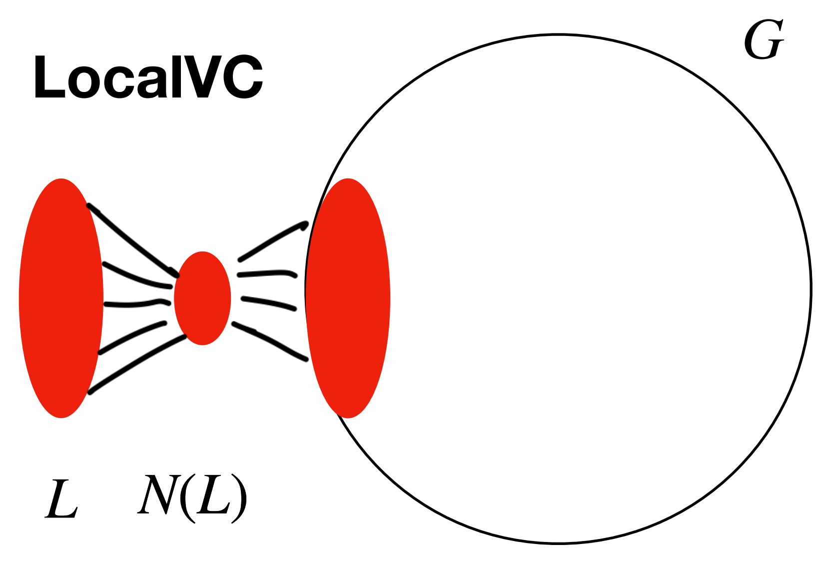
1. Grow DFS tree from *x* using exactly νk edges
2. **If** stuck, **return** a cut from DFS
3. Sample an explore edge (y’,y)
4. Reverse the x,y’-path

Terminate with no cut.



Take Home Message:

"Local algorithms are useful for vertex connectivity"



Open problems

1. **Deterministic** in near-linear time when $m = O(n)$?
   $O(n^{1.8})$-time by [LNSY'19] using expander and page rank.

2. $o(n^3)$ time when $k = \Omega(n)$ or even in **weighted** graph?
   $O(mn)$-time by [Henzinger Rao Gabow'00]

3. $O(m)$ time better **approx**. algorithm?
   $O(\log n)$ approx. by [Censor-Hillel,Ghaffari, and Khun'14]

4. **Local vertex/edge connectivity**
   $O(\nu k)$ time ? More applications?

Questions?

* This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 715672 and 759557. Nanongkai was also partially supported by the Swedish Research Council (Reg. No. 2015-04659.).