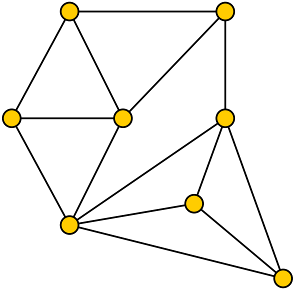
Streaming Algorithms for Independent Sets

Christian Konrad

WARWICK
THE UNIVERSITY OF WARWICK

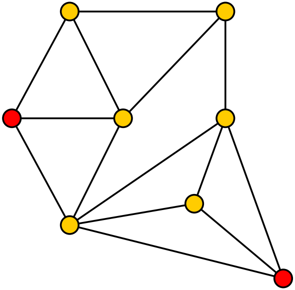
06.06.2017

Independent Sets



Definitions:

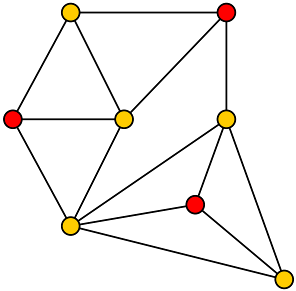
Independent Sets



Definitions:

- Independent Set: Subset of non-adjacent vertices

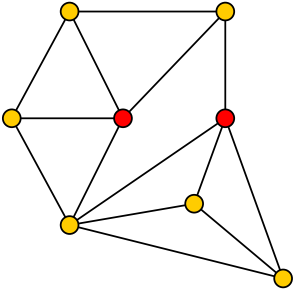
Independent Sets



Definitions:

- Independent Set: Subset of non-adjacent vertices
- Maximum Independent Set: maximum cardinality
  Independent number: $\alpha(G) = 3$

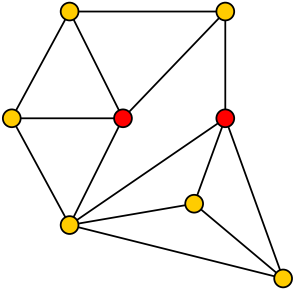
Independent Sets



Definitions:

- Independent Set: Subset of non-adjacent vertices
- Maximum Independent Set: maximum cardinality
  Independent number: $\alpha(G) = 3$
- Maximal Independent Set: cannot be enlarged

Independent Sets



Definitions:

- Independent Set: Subset of non-adjacent vertices
- Maximum Independent Set: maximum cardinality
  Independent number: $\alpha(G) = 3$
- Maximal Independent Set: cannot be enlarged

Applications: Fundamental problem, wireless networks, etc.

Maximum Independent Set

Computational Complexity:

- NP-hard **[Karp, "Reducibility Among Comb. Problems", 1972]**

Maximum Independent Set

Computational Complexity:

- NP-hard **[Karp, "Reducibility Among Comb. Problems", 1972]**
- Hard to Approximate with factor $n^{1-\epsilon}$ **[Håstad, 1999]**

Maximum Independent Set

Computational Complexity:

- NP-hard **[Karp, "Reducibility Among Comb. Problems", 1972]**
- Hard to Approximate with factor $n^{1-\epsilon}$ **[Håstad, 1999]**
- Best Algorithm: $\mathrm{O}(\frac{n \log^2(\log n)}{\log^3 n})$-approximation **[Feige, 2005]**

Maximum Independent Set

Computational Complexity:

- NP-hard **[Karp, "Reducibility Among Comb. Problems", 1972]**
- Hard to Approximate with factor $n^{1-\epsilon}$ **[Håstad, 1999]**
- Best Algorithm: $\mathrm{O}(\frac{n \log^2(\log n)}{\log^3 n})$-approximation **[Feige, 2005]**

Approximation via Maximal Indep. Set?

Maximum Independent Set

Computational Complexity:

- NP-hard **[Karp, "Reducibility Among Comb. Problems", 1972]**
- Hard to Approximate with factor $n^{1-\epsilon}$ **[Håstad, 1999]**
- Best Algorithm: $\mathrm{O}(\frac{n \log^2(\log n)}{\log^3 n})$-approximation **[Feige, 2005]**

Approximation via Maximal Indep. Set? ($\mathrm{O}(d_{\max})$-approximation)

Maximum Independent Set

Computational Complexity:

- NP-hard **[Karp, "Reducibility Among Comb. Problems", 1972]**
- Hard to Approximate with factor $n^{1-\epsilon}$ **[Håstad, 1999]**
- Best Algorithm: $\mathrm{O}(\frac{n \log^2(\log n)}{\log^3 n})$-approximation **[Feige, 2005]**

Approximation via Maximal Indep. Set? ($\mathrm{O}(d_{\max})$-approximation)

How to deal with such a hard problem?

Maximum Independent Set

Computational Complexity:

- NP-hard **[Karp, "Reducibility Among Comb. Problems", 1972]**
- Hard to Approximate with factor $n^{1-\epsilon}$ **[Håstad, 1999]**
- Best Algorithm: $\mathrm{O}(\frac{n \log^2(\log n)}{\log^3 n})$-approximation **[Feige, 2005]**

Approximation via Maximal Indep. Set? ($\mathrm{O}(d_{\max})$-approximation)

How to deal with such a hard problem?

- Specific graph classes: chordal graphs, disc graphs, etc.

Maximum Independent Set

Computational Complexity:

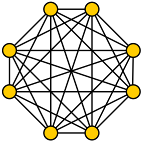
- NP-hard **[Karp, "Reducibility Among Comb. Problems", 1972]**
- Hard to Approximate with factor $n^{1-\epsilon}$ **[Håstad, 1999]**
- Best Algorithm: $\mathrm{O}(\frac{n \log^2(\log n)}{\log^3 n})$-approximation **[Feige, 2005]**

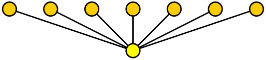
Approximation via Maximal Indep. Set? ($\mathrm{O}(d_{\max})$-approximation)

How to deal with such a hard problem?

- Specific graph classes: chordal graphs, disc graphs, etc.
- *Degree-based quality bounds*

Degree-based Quality Bounds

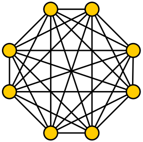


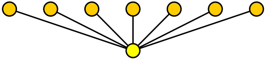


Trivial Bound: $\alpha(G) \geq \frac{n}{d_{\max}+1}$

Good on clique, bad on star

Degree-based Quality Bounds





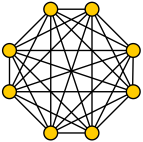
Trivial Bound: $\alpha(G) \geq \frac{n}{d_{\max} + 1}$

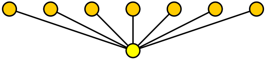
Good on clique, bad on star

Turán Bound: (1941) $\alpha(G) \geq \frac{n}{\overline{d} + 1}$, ($\overline{d}$: avg deg)

Good on a star, bad on clique + star

Degree-based Quality Bounds





Trivial Bound: $\alpha(G) \geq \frac{n}{d_{\max} + 1}$

Good on clique, bad on star

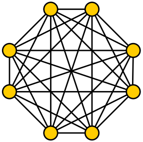
Turán Bound: (1941) $\alpha(G) \geq \frac{n}{\overline{d} + 1}, (\overline{d}:$ avg deg)

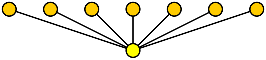
Good on a star, bad on clique + star

Caro-Wei-Bound: (1979/81) $\alpha(G) \geq \beta(G) = \sum_{v \in V} \frac{1}{\deg(v) + 1}$

Good on clique + star

Degree-based Quality Bounds





Trivial Bound: $\alpha(G) \geq \frac{n}{d_{\max}+1}$

Good on clique, bad on star

Turán Bound: (1941) $\alpha(G) \geq \frac{n}{\overline{d}+1}, (\overline{d}: \text{ avg deg})$

Good on a star, bad on clique + star

Caro-Wei-Bound: (1979/81) $\alpha(G) \geq \beta(G) = \sum_{v \in V} \frac{1}{\deg(v)+1}$

Good on clique + star

$$\alpha(G) \geq \sum_{v \in V} \frac{1}{\deg(v)+1} \geq \frac{n}{\overline{d}+1} \geq \frac{n}{d_{\max}+1} .$$

Caro-Wei Bound: Good Bound?

Caro-Wei Bound: $\beta(G) = \sum_{v \in V} \frac{1}{\deg(v)+1}$

Caro-Wei Bound: Good Bound?

Caro-Wei Bound: $\beta(G) = \sum_{v \in V} \frac{1}{\deg(v)+1}$

Greedy Algorithm: achieves the Caro-Wei bound

1. Add vertex of minimum degree to independent set I
2. Remove vertex and its neighborhood from graph
3. Repeat until graph is empty

Caro-Wei Bound: Good Bound?

Caro-Wei Bound: $\beta(G) = \sum_{v \in V} \frac{1}{\deg(v)+1}$

Greedy Algorithm: achieves the Caro-Wei bound

1. Add vertex of minimum degree to independent set I
2. Remove vertex and its neighborhood from graph
3. Repeat until graph is empty

Good on Specific Graph Classes: **[Halldórsson, Konrad, 2015]**

- Caro-Wei bound gives poly-logarithmic approximation ratios on *polynomially bounded-independence graphs*
- On Unit Disc Graphs: $\mathrm{O}((\frac{\log n}{(\log \log n})^2)$-approximation

Caro-Wei Bound: Good Bound?

Caro-Wei Bound: $\beta(G) = \sum_{v \in V} \frac{1}{\deg(v)+1}$

Greedy Algorithm: achieves the Caro-Wei bound

1. Add vertex of minimum degree to independent set I
2. Remove vertex and its neighborhood from graph
3. Repeat until graph is empty

Good on Specific Graph Classes: **[Halldórsson, Konrad, 2015]**

- Caro-Wei bound gives poly-logarithmic approximation ratios on *polynomially bounded-independence graphs*
- On Unit Disc Graphs: $\mathrm{O}((\frac{\log n}{(\log \log n})^2)$-approximation

Efficient Algorithms:

- Distributed Computing: Single round algorithm
- Streaming: One-pass, $\mathrm{O}(n \log n)$ space

Streaming Algorithms for Independent Sets

Streaming Algorithms for Independent Sets

1. **[Halldórsson, Sun, Szegedy, Wang, ICALP 2012]**
   - One-pass C-approximation algorithm using $\tilde{O}(\frac{n^2}{C^2})$ space (Exponential time computations)
   - Every p-pass C-approximation algorithm requires $\Omega(\frac{n^2}{p \cdot C^2})$ space (even for approximating the size only)

Streaming Algorithms for Independent Sets

1. **[Halldórsson, Sun, Szegedy, Wang, ICALP 2012]**
   - One-pass C-approximation algorithm using $\tilde{\mathrm{O}}(\frac{n^2}{C^2})$ space (Exponential time computations)
   - Every p-pass C-approximation algorithm requires $\Omega(\frac{n^2}{p \cdot C^2})$ space (even for approximating the size only)

2. **[Halldórsson, Halldórsson, Losievskaja, Szegedy, ICALP 2010]**
   One-pass, $\mathrm{O}(n \log n)$ space algorithm that outputs independent set of expected size $\beta(G) = \sum_{v \in V} \frac{1}{\deg(v)+1}$

Streaming Algorithms for Independent Sets

1. **[Halldórsson, Sun, Szegedy, Wang, ICALP 2012]**
   - One-pass C-approximation algorithm using $\tilde{O}(\frac{n^2}{C^2})$ space (Exponential time computations)
   - Every p-pass C-approximation algorithm requires $\Omega(\frac{n^2}{p \cdot C^2})$ space (even for approximating the size only)

2. **[Halldórsson, Halldórsson, Losievskaja, Szegedy, ICALP 2010]**
   One-pass, $O(n \log n)$ space algorithm that outputs independent set of expected size $\beta(G) = \sum_{v \in V} \frac{1}{\deg(v)+1}$

3. **[Cormode, Dark, Konrad, 2017, arXiv]**
   One-pass, sublinear space algorithms for approximating the Caro-Wei bound $\sum_{v \in V} \frac{1}{\deg(v)+1}$ itself

Streaming Algorithms for Independent Sets

1. **[Halldórsson, Sun, Szegedy, Wang, ICALP 2012]**
   - One-pass C-approximation algorithm using $\tilde{\mathrm{O}}(\frac{n^2}{C^2})$ space (Exponential time computations)
   - Every p-pass C-approximation algorithm requires $\Omega(\frac{n^2}{p \cdot C^2})$ space (even for approximating the size only)

2. **[Halldórsson, Halldórsson, Losievskaja, Szegedy, ICALP 2010]**
   One-pass, $\mathrm{O}(n \log n)$ space algorithm that outputs independent set of expected size $\beta(G) = \sum_{v \in V} \frac{1}{\deg(v)+1}$

3. **[Cormode, Dark, Konrad, 2017, arXiv]**
   One-pass, sublinear space algorithms for approximating the Caro-Wei bound $\sum_{v \in V} \frac{1}{\deg(v)+1}$ itself

Streaming Algorithm using Exponential Time

Algorithm: C-approximation

1. $V' \leftarrow$ uniform random sample with probability $p = 1/C$
2. Store edges of $G[V']$ while processing the stream
3. Output maximum independent set in $G[V']$

Analysis: Space $\tilde{O}((np)^2) = \tilde{O}(n^2/C^2)$

$$
\begin{aligned}
OPT &= \alpha(G) \\
\mathbb{E}\, ALG &= \mathbb{E}\, \alpha(G[V']) \geq p\alpha(G) \\
OPT/ALG &\geq \frac{1}{p} = C \; .
\end{aligned}
$$

Lower Bound: $\Omega(\frac{n^2}{C^2})$

- Set-Disjointness reduction
- Embedding into *space of edges*

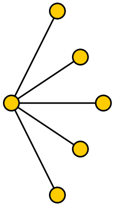
Streaming Algorithms for Independent Sets

1. **[Halldórsson, Sun, Szegedy, Wang, ICALP 2012]**
   - One-pass C-approximation algorithm using $\tilde{\mathrm{O}}(\frac{n^2}{C^2})$ space (Exponential time computations)
   - Every p-pass C-approximation algorithm requires $\Omega(\frac{n^2}{p \cdot C^2})$ space (even for approximating the size only)
2. **[Halldórsson, Halldórsson, Losievskaja, Szegedy, ICALP 2010]**
   One-pass, $\mathrm{O}(n \log n)$ space algorithm that outputs independent set of expected size $\beta(G) = \sum_{v \in V} \frac{1}{\deg(v)+1}$
3. **[Cormode, Dark, Konrad, 2017, arXiv]**
   One-pass, sublinear space algorithms for approximating the Caro-Wei bound $\sum_{v \in V} \frac{1}{\deg(v)+1}$ itself

Streaming Algorithm achieving Caro-Wei Bound

Algorithm:

1. Assign random rank $\in [0,1]$ to every vertex
2. $I \leftarrow V$
3. For each edge uv in stream:
   remove vertex with smaller rank from I
4. return I



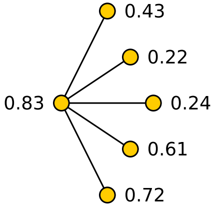
Analysis:

- Vertex survives if and only if it is local maximum
- Vertex v is a local maximum with probability $\frac{1}{\deg(v)+1}$
- $\mathbb{E}\,|I| = \sum_{v \in V} \frac{1}{\deg(v)+1}$.

Streaming Algorithm achieving Caro-Wei Bound

Algorithm:

1. Assign random rank $\in [0, 1]$ to every vertex
2. $I \leftarrow V$
3. For each edge uv in stream:
   remove vertex with smaller rank from I
4. return I



Analysis:

- Vertex survives if and only if it is local maximum
- Vertex v is a local maximum with probability $\frac{1}{\deg(v)+1}$
- $\mathbb{E}\,|I| = \sum_{v \in V} \frac{1}{\deg(v)+1}$.

Streaming Algorithms for Independent Sets

1. **[Halldórsson, Sun, Szegedy, Wang, ICALP 2012]**
    - One-pass C-approximation algorithm using $\tilde{O}(\frac{n^2}{C^2})$ space (Exponential time computations)
    - Every p-pass C-approximation algorithm requires $\Omega(\frac{n^2}{p \cdot C^2})$ space (even for approximating the size only)

2. **[Halldórsson, Halldórsson, Losievskaja, Szegedy, ICALP 2010]**
   One-pass, $O(n \log n)$ space algorithm that outputs independent set of expected size $\beta(G) = \sum_{v \in V} \frac{1}{\deg(v)+1}$

3. **[Cormode, Dark, Konrad, 2017, arXiv]**
   One-pass, sublinear space algorithms for approximating the Caro-Wei bound $\sum_{v \in V} \frac{1}{\deg(v)+1}$ itself

Approximating the Caro-Wei Bound

Problem: Given edge stream, output approximation of $\sum_{v \in V} \frac{1}{\deg(v)+1}$

Starting Point: Frequency Moments

- Degree vector can be seen as a frequency vector f_i
- kth frequency moment:

$$F_k = \sum_i |f_i|^k$$

- Interested in -1-negative frequency moment (harmonic mean)
- **[Braverman, Chestnut, 2015]** If $\sum_i f_i = \Omega(n^2)$ then $\Omega(n)$ space required for $(1+\epsilon)$-approximation to harmonic mean

  (Recall: space roughly $n^{1-2/k}$ is sufficient for positive moments)

Approximating the Caro-Wei Bound

Problem: Given edge stream, output approximation of $\sum_{v \in V} \frac{1}{\deg(v)+1}$

Starting Point: Frequency Moments

- Degree vector can be seen as a frequency vector f_i
- kth frequency moment:

$$F_k = \sum_i |f_i|^k$$

- Interested in -1-negative frequency moment (harmonic mean)
- **[Braverman, Chestnut, 2015]** If $\sum_i f_i = \Omega(n^2)$ then $\Omega(n)$ space required for $(1+\epsilon)$-approximation to harmonic mean

  (Recall: space roughly $n^{1-2/k}$ is sufficient for positive moments)

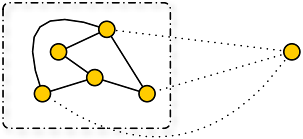
Research Question: Since frequency vector is derived from a graph, can we exploit graph properties to obtain non-trivial algorithms?

Our Results

1. **Edge arrival:** One-pass C-approximation with space $\tilde{O}(\frac{n}{C^2\phi})$, where $\phi \leq \beta(G)$ is a given LB on $\beta(G)$
   Using $\phi = \frac{n}{\overline{d}+1} \leq \beta(G)$, this gives space $\tilde{O}(\frac{\overline{d}}{C^2})$
   poly-log space on graphs with bounded average degree

2. **Vertex arrival:** LB: One-pass C-approx. requires $\Omega(\frac{n}{C^2\beta(G)})$ space
   previous algorithm is optimal

3. **Vertex arrival:** One-pass, $O(\log^3 n)$ space algorithm which outputs a value β' with $\beta' = \Omega(\beta(G)/\log n)$ and $\beta' \leq \alpha(G)$
   different notion of approximation

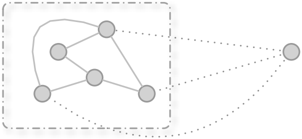


Our Results

1. **Edge arrival:** One-pass C-approximation with space $\tilde{O}(\frac{n}{C^2 \phi})$, where $\phi \leq \beta(G)$ is a given LB on $\beta(G)$
   Using $\phi = \frac{n}{\overline{d}+1} \leq \beta(G)$, this gives space $\tilde{O}(\frac{\overline{d}}{C^2})$
   poly-log space on graphs with bounded average degree

2. **Vertex arrival:** LB: One-pass C-approx. requires $\Omega(\frac{n}{C^2 \beta(G)})$ space
   previous algorithm is optimal

3. **Vertex arrival:** One-pass, $O(\log^3 n)$ space algorithm which outputs a value β' with $\beta' = \Omega(\beta(G)/\log n)$ and $\beta' \leq \alpha(G)$
   different notion of approximation



Algorithm in the Edge-arrival Model

Algorithm 1. Uniform sampling

1. Sample every vertex with prob. p (let V' be set of sampled vertices)
2. Compute $\deg(v)$, for every $v \in V'$
3. Output $\frac{1}{p} \cdot \sum_{v \in V'} \frac{1}{\deg(v)+1}$

Analysis: Standard, e.g. via Chebyshev
Using $p = \frac{1}{C^2}$, we obtain a C-approximation using $\mathrm{O}(n/C^2)$ space

Algorithm in the Edge-arrival Model

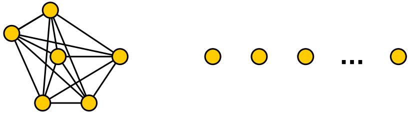
Algorithm 1. Uniform sampling

1. Sample every vertex with prob. p (let V' be set of sampled vertices)
2. Compute $\deg(v)$, for every $v \in V'$
3. Output $\frac{1}{p} \cdot \sum_{v \in V'} \frac{1}{\deg(v)+1}$

Analysis: Standard, e.g. via Chebyshev
Using $p = \frac{1}{C^2}$, we obtain a C-approximation using $\mathrm{O}(n/C^2)$ space

Integrating Dependency on LB $\phi \leq \beta(G)$

Example: $\phi = n^{1/2}$. Suppose G = clique + isolated vertices
$\beta(G)$ = number of isolated vertices $+1$



$p \sim \frac{\log n}{\sqrt{n}}$ is enough!

Algorithm in the Edge-arrival Model (2)

Degree Classes:

- Deg. classes $V = V_1, \ldots, V_{\log_c n}$ with $v \in V_i$ iff $c^i \leq \deg(v) < c^{i+1}$
- *Heavy degree class*: $|V_i| \geq \frac{\beta(G)}{g \cdot \log_c n}$, for some $g \geq 1$
  Light degree classes can be ignored
- Set p to roughly $\frac{\text{poly}\log(n) \cdot g}{\beta(G)}$. This ensures that $\log n$ vertices of each heavy degree class are sampled (concentration!)

Algorithm in the Edge-arrival Model (2)

Degree Classes:

- Deg. classes $V = V_1, \dots, V_{\log_c n}$ with $v \in V_i$ iff $c^i \leq \deg(v) < c^{i+1}$
- *Heavy degree class*: $|V_i| \geq \frac{\beta(G)}{g \cdot \log_c n}$, for some $g \geq 1$
  Light degree classes can be ignored
- Set p to roughly $\frac{\mathrm{poly}\log(n) \cdot g}{\beta(G)}$. This ensures that $\log n$ vertices of each heavy degree class are sampled (concentration!)

Algorithm 2. Uniform sampling with degree classes

1. Sample every vertex with prob. p (let V' be set of sampled vertices)
2. Compute $\deg(v)$, for every $v \in V'$
3. Partition sampled vertices into degree classes $V'_1, V'_2, \dots$
4. Only make V'_i contribute to estimator if $|V'_i| = \Omega(\log n)$

Algorithm in the Edge-arrival Model (2)

Degree Classes:

- Deg. classes $V = V_1, \dots, V_{\log_c n}$ with $v \in V_i$ iff $c^i \leq \deg(v) < c^{i+1}$
- *Heavy degree class*: $|V_i| \geq \frac{\beta(G)}{g \cdot \log_c n}$, for some $g \geq 1$
  Light degree classes can be ignored
- Set p to roughly $\frac{\mathrm{poly}\log(n) \cdot g}{\beta(G)}$. This ensures that $\log n$ vertices of each heavy degree class are sampled (concentration!)

Algorithm 2. Uniform sampling with degree classes

1. Sample every vertex with prob. p (let V' be set of sampled vertices)
2. Compute $\deg(v)$, for every $v \in V'$
3. Partition sampled vertices into degree classes $V'_1, V'_2, \dots$
4. Only make V'_i contribute to estimator if $|V'_i| = \Omega(\log n)$

Theorem:

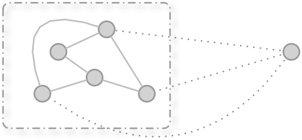
- C-approximation with $\tilde{\mathrm{O}}(\frac{n}{C^2 \phi})$ space
- $(1+\epsilon)$-approximation with $\tilde{\mathrm{O}}(\frac{n}{\mathrm{poly}(\epsilon)\phi})$ space

Our Results

1. **Edge arrival:** One-pass C-approximation with space $\tilde{O}(\frac{n}{C^2 \phi})$, where $\phi \leq \beta(G)$ is a given LB on $\beta(G)$
   Using $\phi = \frac{n}{\overline{d}+1} \leq \beta(G)$, this gives space $\tilde{O}(\frac{\overline{d}}{C^2})$
   poly-log space on graphs with bounded average degree

2. **Vertex arrival:** LB: One-pass C-approx. requires $\Omega(\frac{n}{C^2 \beta(G)})$ space
   previous algorithm is optimal

3. **Vertex arrival:** One-pass, $O(\log^3 n)$ space algorithm which outputs a value β' with $\beta' = \Omega(\beta(G)/\log n)$ and $\beta' \leq \alpha(G)$
   different notion of approximation



Space Lower Bound

Set-Disjointness: X, Y disjoint?

$$\begin{array}{ccc} \text{Alice} & \longleftrightarrow & \text{Bob} \\ X \subseteq [k] & & Y \subseteq [k] \end{array}$$

Space Lower Bound

Set-Disjointness: X, Y disjoint?

$$\begin{array}{ccc} \text{Alice} & \longleftrightarrow & \text{Bob} \\ X \subseteq [k] & & Y \subseteq [k] \end{array}$$

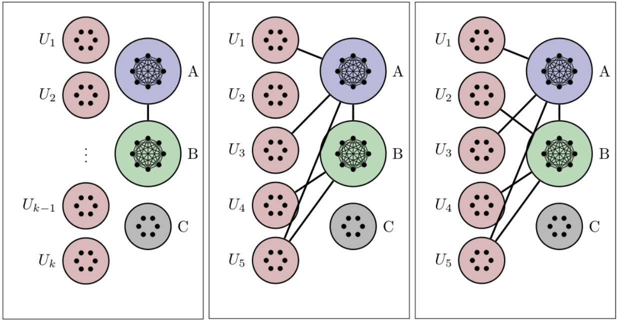
Theorem: $R(\textsc{Disj}_k) = \Omega(k)$

Space Lower Bound

Set-Disjointness: X, Y disjoint?

Alice	$\longleftrightarrow$	Bob
$X \subseteq [k]$		$Y \subseteq [k]$

Theorem: $R(\text{DISJ}_k) = \Omega(k)$



(a) Initial configuration. **(b)** Example with $X = \{2,4\}$ and $Y = \{1,2,3\}$. **(c)** Example with $X = \{2,4\}$ and $Y = \{1,3\}$.

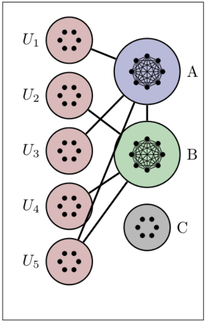
Space Lower Bound (2)

Reduction:

- $X \cap Y = \varnothing$: Left vertices have large degrees
  Caro-Wei bound is small
- $X \cap Y \neq \varnothing$: Intersection vertices isolated
  Caro-Wei bound is large

Vertex-arrival Order: $C, U_1, \dots, U_k, A, B$

Theorem: Every one-pass streaming algorithm that approximates $\beta(G)$ within a factor of C requires $\Omega(\frac{n}{C^2\beta(G)})$ space



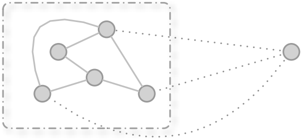
(c) Example with $X = \{2, 4\}$ and $Y = \{1, 3\}$.

Our Results

1. **Edge arrival:** One-pass C-approximation with space $\tilde{O}(\frac{n}{C^2\phi})$, where $\phi \leq \beta(G)$ is a given LB on $\beta(G)$
   Using $\phi = \frac{n}{\overline{d}+1} \leq \beta(G)$, this gives space $\tilde{O}(\frac{\overline{d}}{C^2})$
   poly-log space on graphs with bounded average degree

2. **Vertex arrival:** LB: One-pass C-approx. requires $\Omega(\frac{n}{C^2\beta(G)})$ space
   previous algorithm is optimal

3. **Vertex arrival:** One-pass, $O(\log^3 n)$ space algorithm which outputs a value β' with $\beta' = \Omega(\beta(G)/\log n)$ and $\beta' \leq \alpha(G)$
   different notion of approximation



Vertex-arrival Order

Notation:

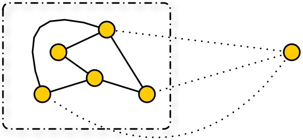
- Arrival Order: $v_1, v_2, v_3, \ldots, v_n$
- $V^k = \cup_{j \leq k} \{v_j\}$, $G^k = G[V^k]$, subgraph seen in round k
- $V_i^k = \{v \in V^k \ : \ \deg_{G^k}(v) \leq 2^i\}$, nodes of degree at most 2^i in G^k

Approximation:

$$\max_{1 \leq i \leq \lceil \log n \rceil} |V_i^k| = \Omega(\beta(G^k)/\log n)$$

Monotonicity:

- Independence nbr: $\alpha(G^k) \leq \alpha(G^{k+1})$
- (Caro-Wei Bound: $\beta(G^k) \leq \beta(G^{k+1})$ does not hold!)



Task: For every $1 \leq i \leq \lceil \log n \rceil$, approximate:

$$\max_k |V_i^k|$$

Vertex-arrival Order (2)

Algorithm: input i

1. Sample $S \leftarrow \varnothing$
2. Sampling probability $p \leftarrow 1$
3. While stream not empty ($v \leftarrow$ next vertex in stream)
    1. Add v to sample S if $\deg_{G^k}(v) \leq 2^i$
    2. Remove vertices u from S with $\deg_{G^k}(u) > 2^i$
    3. If $|S| \geq c \log n$ then downsample as follows:
       Remove every vertex from S with probability $1/2$, update $p = p/2$
4. Return $c \log n / p$

Properties:

- Conditioned on p, S is a uniform sample of V_i^k
- Space $\mathrm{O}(\log^2 n)$ (store $\log n$ nodes and their degrees)

Theorem: One-pass, $\mathrm{O}(\log^3 n)$ space algorithm which outputs a value β' with $\beta' = \Omega(\beta(G)/\log n)$ and $\beta' \leq \alpha(G)$

Conclusion

Results:

- Tight approximation algorithm for $\beta(G)$ in the edge arrival model
- Vertex arrival model algorithm

Conclusion

Results:

- Tight approximation algorithm for $\beta(G)$ in the edge arrival model
- Vertex arrival model algorithm

Future Work and Preliminary Results: Maximal Independent Set

- One-pass: $\Omega(n^2)$ space lower bound
- $\mathrm{O}(\log n)$ pass upper bound (via Luby's algorithm from distributed computing)
- Prove space/passes trade-off ?

Conclusion

Results:

- Tight approximation algorithm for $\beta(G)$ in the edge arrival model
- Vertex arrival model algorithm

Future Work and Preliminary Results: Maximal Independent Set

- One-pass: $\Omega(n^2)$ space lower bound
- $\mathrm{O}(\log n)$ pass upper bound (via Luby's algorithm from distributed computing)
- Prove space/passes trade-off ?

Thank you