Limits of Local Algorithms in Random Graphs

Madhu Sudan
MSR

Joint work with David Gamarnik (MIT)
Preliminaries

- **Terminology:**
  - Graph $G = (V, E)$; $E \subseteq V \times V$ symmetric
  - $V$: Vertices; $E$: edges;
  - Independent Set: $I \subseteq V$ s.t. $\{u, v\} \subseteq I \Rightarrow (u, v) \notin E$.

- **Algorithmic Challenge:** Given $G$, find large independent set $I$.
- **[Karp’72]:** NP-complete in worst-case.
Random Graphs

• Popularized by Erdös-Renyi:
  – Basic Model: Every edge thrown in independently with probability \( p \).
  – Regular Model: Pick \( G \) uniformly among all \( d \)-regular graphs:
    • \( d \)-regular: Every vertex in exactly \( d \) edges.

• Background: Almost surely, random \( d \)-regular graph on \( n \) vertices has independent set of size
  \( (1 + o(1)) \cdot c_d \cdot n \) for \( c_d = \frac{2}{d} \log d \).

• Question: Find such large independent sets?
Random Graphs & Complexity

• Worst-case complexity results no longer apply.
• Could hope: Some polynomial time algorithm finds ind. sets of size \((1 - o(1)) \cdot c_d \cdot n\)

• Greedy algorithm:
  – Order vertices arbitrarily.
  – Run through vertices in order, include \(v\) in \(I\) if this keeps \(I\) independent.

• Fact: Finds set of size \(\approx c_d \cdot \frac{n}{2}\)
Main Result

• Our Theorem: “Local algorithms” can not. In fact they fall short by a constant factor.

• Extensions/Subsequent results:
  – [Rahman-Virag]: Fall short by factor of $\frac{1}{2}$.
  – Locally-guided decimation algorithms (Belief Propagation, Survey Propagation) fail on some other CSPs.
Definition: Local Algorithms

• Informally: Local algorithms
  – Input = Communication network.
  – Wish to use local communication to compute some property of input.
  – In our case – large independent set in graph.
  – Allowed to use randomness, generated locally.
Formally

- (Randomized) Decision Algorithm:
  - $f(u, G, \vec{w}) \in \{0,1\}$: Determines if $u \in I$.
    - $\vec{w}$ is a weighting, say in $[0,1]$, on vertices
- Correctness:
  - $\forall u, v, G, \vec{w}$ s.t. $u \leftrightarrow_G v$,
    $$f(u, G, w) = 0 \text{ or } f(v, G, w) = 0.$$
- Locality:
  - $f$ is $r$-local if $f(u, G, \vec{w}) = f(v, H, \vec{x})$ whenever $r$-local weighted neighborhood around $u$ in $(G, \vec{w})$ and $v$ in $(H, \vec{x})$ are identical.
Locality ≠ Locality

• Locality in distributed algorithms
  – Usually algorithms try to compute some function of input graph, on the graph itself.
  – Algorithm uses data available topologically locally.
  – Leads to our model

• Locality a la Codes/Property Testing
  – Locality simply refers to number of queries to input.
  – More general model.
  – We can’t/don’t deal with it.
Motivations for our work

1. Paucity of “complexity” results for random graphs. Major exceptions:
   - Rossman: $AC^0$/Monotone complexity of planted clique.
   - Feige-Krauthgamer: LP relaxations.

2. Physicists explanation of complexity
   - Clustering/Shattering explain inability of algorithms.

3. Graph Limit theory
   - Local characteristics of (random) graphs predict global properties (nearly).
Motivations (contd.)

• Specific conjecture [Hatami-Lovasz-Szegedy]:
  As $r \to \infty$, $r$-local algorithms should find independent sets of cardinality $c_d(1 - o(1))n$.

• Refuted by our theorem.
Proof

• Part I:
  – A clustering phenomenon for independent sets in random graphs [Inspired by Coja-Oglan].

• Part II:
  – Locality ⇒ Continuity ⇒ ¬(Clustering).

Both parts simple.
Clustering Phenomena

• Generally:
  – When you look at “near-optimal” solutions, then they are very structured.
  – $\Rightarrow$ topology of solutions highly disconnected (in Hamming space).

• In our context
  – Consider graph on independent sets (of size $\approx c_d n$) with $I \leftrightarrow J$ if $|I \Delta J| \leq \epsilon \cdot n$.
  – Highly disconnected?
Clustering Theorem

• Theorem: $\forall d, \exists 0 < \theta < \tau < c_d$ s.t.:
  - Almost surely over $G$, $\forall I, J$ of size $\approx c_d n$,
    $\frac{|I \cap J|}{n} \notin (\theta, \tau)$

• Proof:
  - Compute expected number of independent sets with forbidden intersection and note it is $\ll 1$.
  - Second moment proves concentration.

• Implies Clustering.
Locality $\Rightarrow \neg$(Clustering)

• Main Idea:
  – Fix $r$-local function $f$, that usually produces independent sets of size $\approx c_d \cdot n$
  – Sample weights twice: $\vec{w}$, and then $\vec{x}$; $p$-correlatedly.
  – Let $I = f(G, \vec{w})$ and $J = f(G, \vec{x})$.
  – Prove:
    • whp, $|I|, |J| \approx c_d \cdot n$
    • whp, $|I \cap J| \approx \beta(p) \cdot n$
    • $\exists p$ s.t. $\beta(p) \in (\theta, \tau)$
Size of Ind. Set

• Claim: Size of independent set produced by local algorithms is concentrated.
  – Let $\alpha = \alpha(f) = \mathbb{E}_{\overrightarrow{w}}[f(u, \mathbb{T}_d, \overrightarrow{w})]$ (where $\mathbb{T}_d = \text{infinite tree of degree } d$)
  – W.p. 1-o(1), size of ind. set produced $\approx \alpha \cdot n$.

• Proof:
  – Most neighborhoods are trees $\Rightarrow$ Expectation.
  – Most neighborhoods are disjoint $\Rightarrow$ Chebychev.
$p$-correlated distributions

- Pick $\vec{w}, \vec{y} \in [0,1]^n$, independently.
- Let $\vec{x}_i = \vec{w}_i$ w.p. $p$ and $\vec{y}_i$ otherwise, independently for each $i$.
- Let $\beta(p) = \mathbb{E}_{\vec{w}, \vec{x}}[f(u, T_d, \vec{w}) \land f(u, T_d, \vec{x})]$.
- As in previous argument:
  - $\mathbb{E}[|I \cap J|] \approx \beta(p) \cdot n$
  - $|I \cap J|$ concentrated around expectation.
Continuity of $\beta(p)$

- Fix $\vec{w}, \vec{y}$, and consider
  \[ \Pr[f(u, T_d, \vec{w}) \land f(u, T_d, \vec{x})] \]
- Above expression is some polynomial in $p$, of degree at most $d^r$.
- In particular, it is continuous as function of $p$.
- $\Rightarrow \beta(p) =$ Expectation over $\vec{w}, \vec{y}$ is also continuous.
- Suffices to show $[\beta(0), \beta(1)] \cap (\theta, \tau) \neq \emptyset$. 
Continuity (contd.)

- \( \beta(p) = \mathbb{E}_{\bar{w}, \bar{x}}[f(u, T_d, \bar{w}) \land f(u, T_d, \bar{x})] \)
- \( \beta(1) = \alpha(f) \approx c_d \)
- \( \beta(0) = \alpha^2 \approx c_d^2 \)
- Follows from calculations (also naturally) that
  - \([\beta(0), \beta(1)] \cap (\theta, \tau) \neq \emptyset\)
- Conclude:
  - w.h.p., \(|I|, |J| \approx c_d \cdot n\)
  - w.h.p., \(|I \cap J| \approx \beta(p) \cdot n\)
  - \(\exists p \text{ s.t. } \beta(p) \in (\theta, \tau)\)
Extensions-1

• Our notion of clustering:
  – ∀I,J independent: |I|, |J| \approx \alpha c_d n, |I \cap J| \notin (\theta \cdot n, \tau \cdot n)
  – To get \theta < \tau, need \alpha close to 1.

• To improve [Ramzan-Virag] suggest:
  – ∀I_1, I_2, ..., I_m with |I_j| \approx \alpha c_d n, \exists i, j s.t.
    |I_i \cap I_j| \notin (\theta \cdot n, \tau \cdot n)
  – Lets them get to \alpha \rightarrow \frac{1}{2}
Extensions-2

• Local algorithms: Makes all decisions locally, in one shot.

• Locally guided decimation algorithms:
  – Compute some local information.
  – Make one decision (e.g., $\nu \in I$?) and commit
  – Repeat.

• Recent work: Locally guided decimation algorithms also don’t get close to optimum (on other random CSPs).
Conclusions

• “Clustering” is an obstacle?

• Answer:
  – At least to local algorithms.
  – Local algorithms behave continuously, forcing non-clustering of solutions.

• Open questions:
  – Barrier to local algorithms in general sense?
  – To other complexity classes?
Thank You