Robust Shattering Arguments (part II)

Alexandre Nolin

Télécom SudParis, Institut Polytechnique de Paris

Warwick's Center in Venice, 20/05/2026

joint work with

Mohsen Ghaffari
Magnús M. Halldórsson
Yannic Maus

With inputs from

Sebastian Brandt
Christoph Grunau
Václav Rozhoň

Topic of the talk

Two kinds of talk

Two kinds of talk

↑ today ↑

Our topic

Well known result by Fischer and Ghaffari

Theorem ([FG17])

There is an algorithm of complexity $O(d^2 + T_{\mathrm{LLL}}^{\mathsf{det}}(\operatorname{poly}\log n))$ *for the constructive Lovász Local Lemma under criterion* $8pd^9 < 1$ *in the randomized LOCAL model.*

Our topic

Well known result by Fischer and Ghaffari

Theorem ([FG17])

There is an algorithm of complexity $O(d^2 + T^{\det}_{\mathrm{LLL}}(\operatorname{poly}\log n))$ *for the constructive Lovász Local Lemma under criterion* $8pd^9 < 1$ *in the randomized LOCAL model.*

For the value of $T^{\det}_{\mathrm{LLL}}(\operatorname{poly}\log n) \in O(\operatorname{poly}\log\log n)$ see [MT10, CPS17, GHK18]

Our topic

Well known result by Fischer and Ghaffari

Theorem ([FG17])

There is an algorithm of complexity $O(d^2 + T^{\text{det}}_{\mathrm{LLL}}(\operatorname{poly}\log n))$ for the constructive Lovász Local Lemma under criterion $8pd^9 < 1$ in the randomized LOCAL model.

For the value of $T^{\text{det}}_{\mathrm{LLL}}(\operatorname{poly}\log n) \in O(\operatorname{poly}\log\log n)$ see [MT10, CPS17, GHK18]

decomposing the complexity:

$O(d^2)$: pre-shattering

$T^{\text{det}}_{\mathrm{LLL}}(\operatorname{poly}\log n))$: post-shattering

Our topic

Well known result by Fischer and Ghaffari

Theorem ([FG17])

There is an algorithm of complexity $O(d^2 + T_{\mathrm{LLL}}^{\mathrm{det}}(\operatorname{poly}\log n))$ for the constructive Lovász Local Lemma under criterion $8pd^9 < 1$ in the randomized LOCAL model.

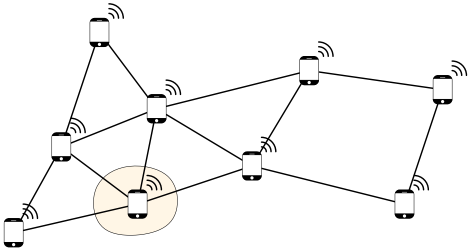
For the value of $T_{\mathrm{LLL}}^{\mathrm{det}}(\operatorname{poly}\log n) \in O(\operatorname{poly}\log\log n)$ see [MT10, CPS17, GHK18]

decomposing the complexity:

$O(d^2)$: **pre-shattering** $T_{\mathrm{LLL}}^{\mathrm{det}}(\operatorname{poly}\log n))$: post-shattering

Problem definition

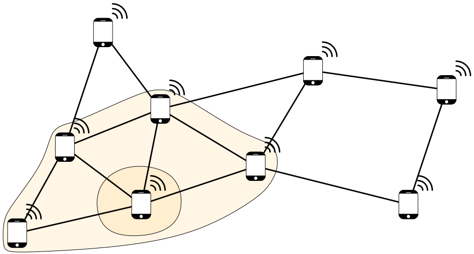
LOCAL model



- synchronous,
- messages of unbounded size,
- no limit on computational power,
- nodes given unique identifiers, between 1 and poly(n)
- polynomial upper bound on n known by the nodes (n: size of the graph)

Goal: minimizing the number of communication rounds (time unit).

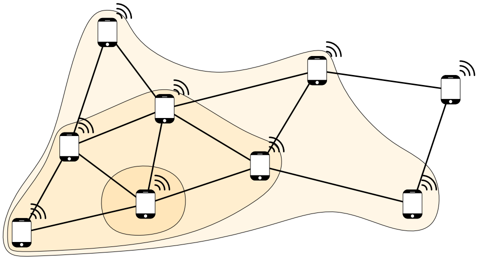
LOCAL model



- synchronous,
- messages of unbounded size,
- no limit on computational power,
- nodes given unique identifiers, between 1 and poly(n)
- polynomial upper bound on n known by the nodes (n: size of the graph)

Goal: minimizing the number of communication rounds (time unit).

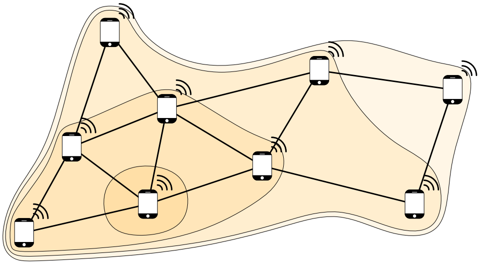
LOCAL model



- synchronous,
- messages of unbounded size,
- no limit on computational power,
- nodes given unique identifiers, between 1 and poly(n)
- polynomial upper bound on n known by the nodes (n: size of the graph)

Goal: minimizing the number of communication rounds (time unit).

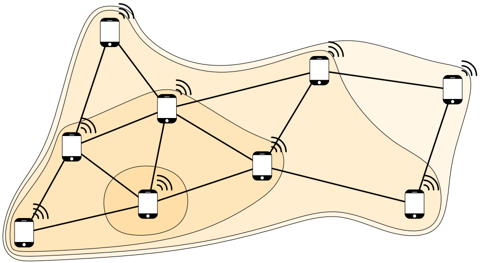
LOCAL model



- synchronous,
- messages of unbounded size,
- no limit on computational power,
- nodes given unique identifiers, between 1 and poly(n)
- polynomial upper bound on n known by the nodes (n: size of the graph)

Goal: minimizing the number of communication rounds (time unit).

LOCAL model

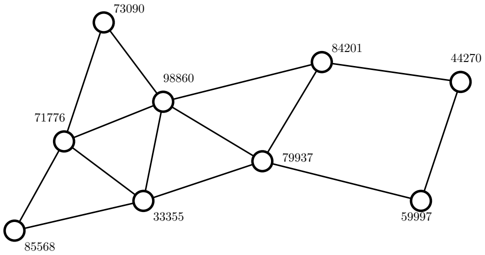


- synchronous,
- messages of unbounded size,
- no limit on computational power,
- nodes given unique identifiers, between 1 and poly(n)
- polynomial upper bound on n known by the nodes (n: size of the graph)

Goal: minimizing the number of communication rounds (time unit).
Randomized variant: allowed to fail with probability $1/n$

What is LLL?

Answer: A kind of probabilistic method

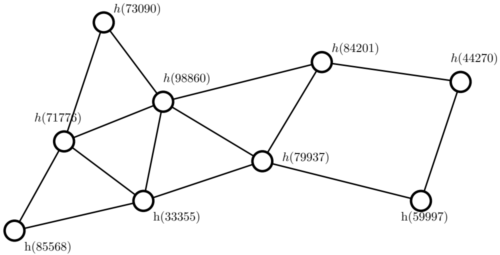


Graph G where nodes have huge identifier (up to 2^n).

Want to update $h(x) = x \mod p$ everywhere without having a collision.

What is LLL?

Answer: A kind of probabilistic method

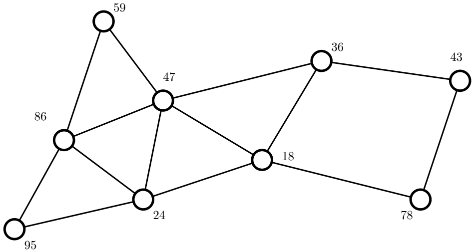


Graph G where nodes have huge identifier (up to 2^n).

Want to update $h(x) = x \mod p$ everywhere without having a collision.

What is LLL?

Answer: A kind of probabilistic method

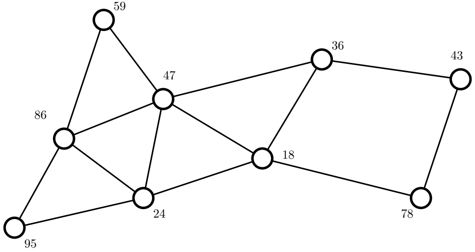


Graph G where nodes have huge identifier (up to 2^n).

Want to update $h(x) = x \mod p$ everywhere without having a collision.

What is LLL?

Answer: A kind of probabilistic method



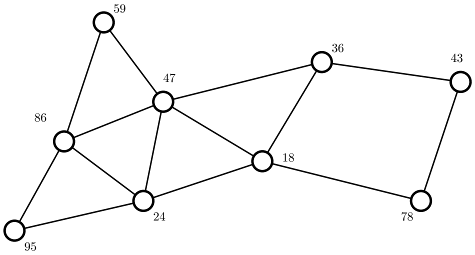
Graph G where nodes have huge identifier (up to 2^n).

Want to update $h(x) = x \mod p$ everywhere without having a collision.

Pick a random prime $p \in [1, 10n^2 \log n]$. Probability that nodes u and v get the same ID ("bad event"): $< n^{-2}$

What is LLL?

Answer: A kind of probabilistic method



Graph G where nodes have huge identifier (up to 2^n).

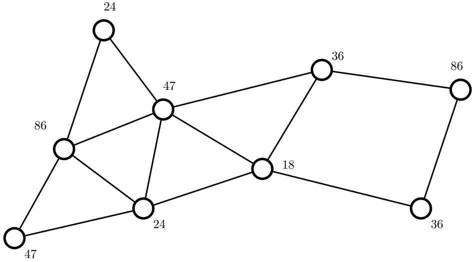
Want to update $h(x) = x \mod p$ everywhere without having a collision.

Pick a random prime $p \in [1, 10n^2 \log n]$. Probability that nodes u and v get the same ID ("bad event"): $< n^{-2}$

There are $\binom{n}{2}$ pairs of nodes: $\binom{n}{2} \cdot n^{-2} < 1$, so there exists p s.t. all $h(x)$ differ.

What is LLL? cont.

What if: bad events are more likely, but only local?

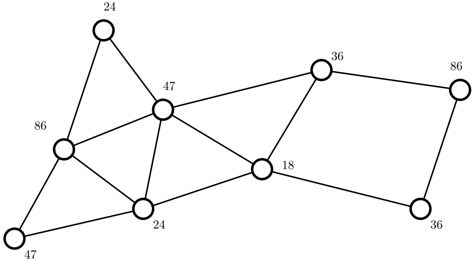


Want: coloring, each color repeats $\leq 99 \log \Delta$ times around each node.

Every node just picks a random color in $[C] = [\Delta / \log \Delta]$, where Δ=max degree.

What is LLL? cont.

What if: bad events are more likely, but only local?



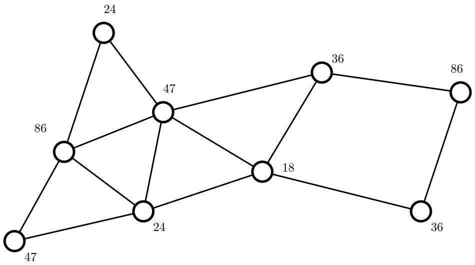
Want: coloring, each color repeats $\leq 99 \log \Delta$ times around each node.

Every node just picks a random color in $[C] = [\Delta / \log \Delta]$, where Δ=max degree.

Bad event at each node: a color repeats $> 99 \log \Delta$ times.
Prob. of bad event: $< C \cdot \exp(-\Omega(\log \Delta)) < \Delta^{-20}$

What is LLL? cont.

What if: bad events are more likely, but only local?



Want: coloring, each color repeats $\leq 99 \log \Delta$ times around each node.

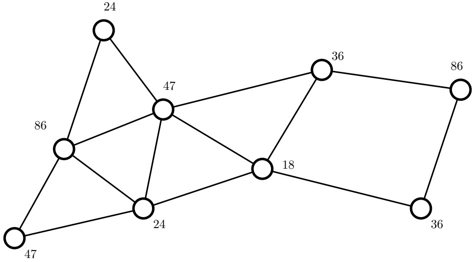
Every node just picks a random color in $[C] = [\Delta / \log \Delta]$, where Δ=max degree.

Bad event at each node: a color repeats $> 99 \log \Delta$ times.
Prob. of bad event: $< C \cdot \exp(-\Omega(\log \Delta)) < \Delta^{-20}$

$n \cdot \Delta^{-20} >> 1$: union bound does not work

What is LLL? cont.

What if: bad events are more likely, but only local?



Want: coloring, each color repeats $\leq 99 \log \Delta$ times around each node.

Every node just picks a random color in $[C] = [\Delta/\log \Delta]$, where Δ=max degree.

Bad event at each node: a color repeats $> 99 \log \Delta$ times.
Prob. of bad event: $< C \cdot \exp(-\Omega(\log \Delta)) < \Delta^{-20}$

$n \cdot \Delta^{-20} >> 1$: union bound does not work

But: every event is only non-independent with $\leq \Delta^2$ other events.

Lovász Local Lemma (symmetric version)

Assume:

- Set of "bad events" $\mathcal{E}$,
- $p \in [0, 1]$ s.t.: Each event has probability $\leq p$,
- Dependency graph: an edge between two events if non-independent,
- d: max degree of dependency graph.

Theorem (LLL)

If $ep(d+1) < 1$*, with probability* > 0*, no bad event occurs.*

Lovász Local Lemma (symmetric version)

Assume:

- Set of "bad events" $\mathcal{E}$,
- $p \in [0, 1]$ s.t.: Each event has probability $\leq p$,
- Dependency graph: an edge between two events if non-independent,
- d: max degree of dependency graph.

Theorem (LLL)

If $ep(d+1) < 1$, with probability > 0, no bad event occurs.

Constructive LLL: finding an assignment of random variables s.t. no bad event occurs.

Importance of LLL

LLL complete for LCLs of complexity $o(\log n)$ in randomized LOCAL [CP19]

Several problems are easily cast / reduced as solving an LLL: sinkless orientation, splitting problems, edge coloring problems...

Importance of LLL

LLL complete for LCLs of complexity $o(\log n)$ in randomized LOCAL [CP19]

Several problems are easily cast / reduced as solving an LLL: sinkless orientation, splitting problems, edge coloring problems...

For many problems, LLL is the only reason that we know that a solution even exists: we have no other approach!

Shattering LLL

Fischer-Ghaffari result

Theorem ([FG17])

There is an algorithm of complexity $O(d^2 + T^{\text{det}}_{\mathrm{LLL}}(\text{poly}\log n))$ for the constructive Lovász Local Lemma under criterion $8pd^9 < 1$ in the randomized LOCAL model.

Fischer-Ghaffari result

Theorem ([FG17])

There is an algorithm of complexity $O(d^2 + T_{\mathrm{LLL}}^{\mathrm{det}}(\mathrm{poly}\log n))$ for the constructive Lovász Local Lemma under criterion $8pd^9 < 1$ in the randomized LOCAL model.

$O(d^2)$: pre-shattering

$T_{\mathrm{LLL}}^{\mathrm{det}}(\mathrm{poly}\log n)$: post-shattering

Fischer-Ghaffari result

Theorem ([FG17])

There is an algorithm of complexity $O(d^2 + T_{\mathrm{LLL}}^{\mathrm{det}}(\operatorname{poly}\log n))$ for the constructive Lovász Local Lemma under criterion $8pd^9 < 1$ in the randomized LOCAL model.

$O(d^2)$: pre-shattering

$T_{\mathrm{LLL}}^{\mathrm{det}}(\operatorname{poly}\log n)$: post-shattering

What could shattering look like for LLL?

Recall: valid shattering lemma

Lemma

If for every x-independent set S, the probability that all nodes in S join a "post-shattering" set B is at most $\Delta^{-(x+4)|S|}$, then w.h.p., B contains no connected component of size $\geq \Delta^x \log n$

We have three tasks:

Recall: valid shattering lemma

Lemma

If for every x-independent set S, the probability that all nodes in S join a "post-shattering" set B is at most $\Delta^{-(x+4)|S|}$*, then w.h.p., B contains no connected component of size* $\geq \Delta^x \log n$

We have three tasks:

1. solving most of the instance (probability of a single node surviving is small)

Recall: valid shattering lemma

Lemma

If for every x-independent set S, the probability that all nodes in S join a "post-shattering" set B is at most $\Delta^{-(x+4)|S|}$*, then w.h.p., B contains no connected component of size* $\geq \Delta^x \log n$

We have three tasks:

1. solving most of the instance (probability of a single node surviving is small)
2. a set of distant nodes should become solved roughly independently (to apply above lemma)

Recall: valid shattering lemma

Lemma

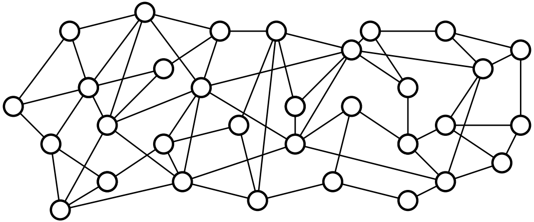
If for every x-independent set S, the probability that all nodes in S join a "post-shattering" set B is at most $\Delta^{-(x+4)|S|}$*, then w.h.p., B contains no connected component of size* $\geq \Delta^x \log n$

We have three tasks:

1. solving most of the instance (probability of a single node surviving is small)
2. a set of distant nodes should become solved roughly independently (to apply above lemma)
3. what remains unsolved should be solvable, and fast (in our case: it's still an LLL)

What happens if we just sample everything?

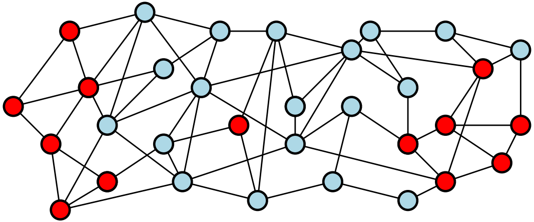
Consider a set S of bad events, at pairwise distance ≥ 2 in the dependency graph.
Prob. that they all hold: $< p^{|S|}$.



Claim: bad events that hold induce small connected components

What happens if we just sample everything?

Consider a set S of bad events, at pairwise distance ≥ 2 in the dependency graph.
Prob. that they all hold: $< p^{|S|}$.

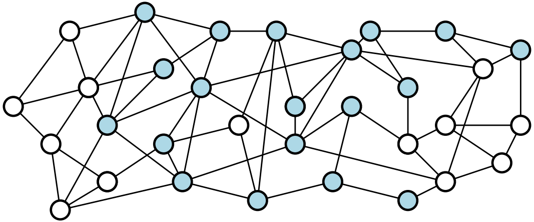


Claim: bad events that hold induce small connected components

Issue: no reason to believe that one can fix the partial solution purely within these components.

What happens if we just sample everything?

Consider a set S of bad events, at pairwise distance ≥ 2 in the dependency graph.
Prob. that they all hold: $< p^{|S|}$.

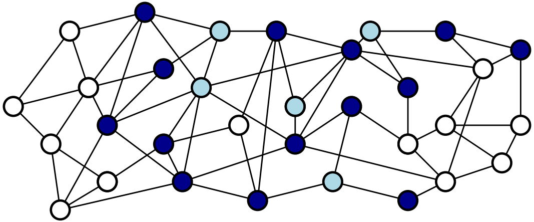


Claim: bad events that hold induce small connected components

Issue: no reason to believe that one can fix the partial solution purely within these components.

What happens if we just sample everything?

Consider a set S of bad events, at pairwise distance ≥ 2 in the dependency graph. Prob. that they all hold: $< p^{|S|}$.

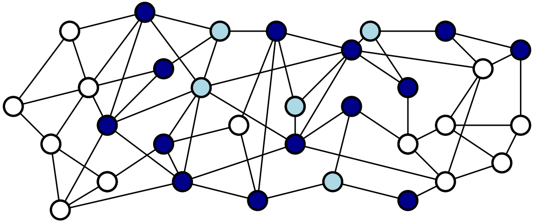


Claim: bad events that hold induce small connected components

Issue: no reason to believe that one can fix the partial solution purely within these components.

What happens if we just sample everything?

Consider a set S of bad events, at pairwise distance ≥ 2 in the dependency graph.
Prob. that they all hold: $< p^{|S|}$.



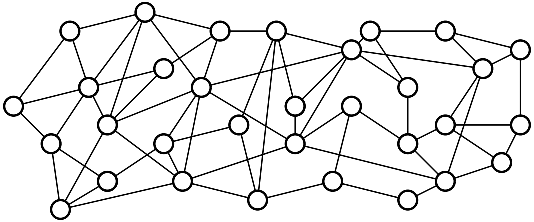
Claim: bad events that hold induce small connected components

Issue: no reason to believe that one can fix the partial solution purely within these components.

And if resampling: some events might need $\Omega(\log_p n)$ resamplings

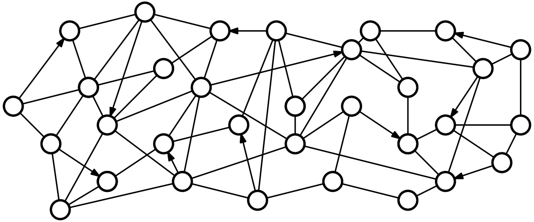
Hint from sinkless orientation

Sample edges, but not ALL edges [GS17].



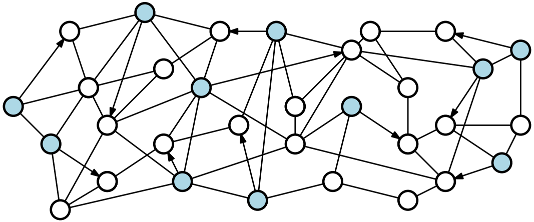
Hint from sinkless orientation

Sample edges, but not ALL edges [GS17].



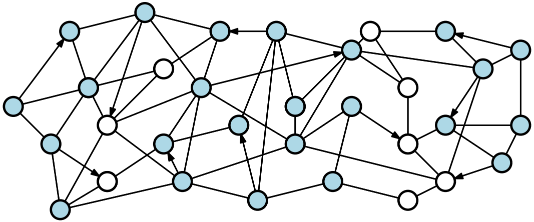
Hint from sinkless orientation

Sample edges, but not ALL edges [GS17].



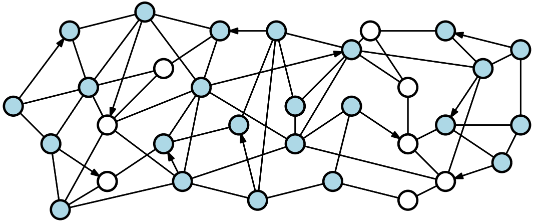
Hint from sinkless orientation

Sample edges, but not ALL edges [GS17].



Hint from sinkless orientation

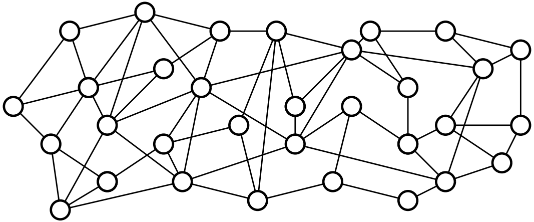
Sample edges, but not ALL edges [GS17].



For LLL: sample many events and variables, but not ALL events and variables.

Freezing variables

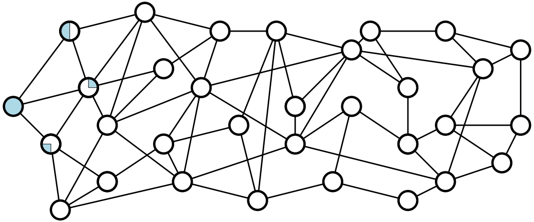
Idea: progressively sample, stop when marginal probability becomes too high [MR98].



Fix threshold $q < p$. If marginal probability of an event reaches q, stop sampling its variables.

Freezing variables

Idea: progressively sample, stop when marginal probability becomes too high [MR98].

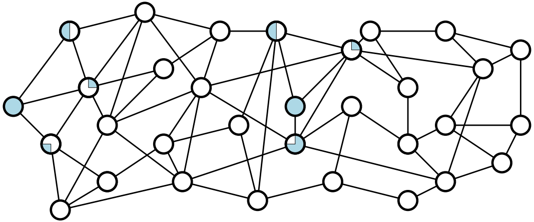


Fix threshold $q < p$. If marginal probability of an event reaches q, stop sampling its variables.

Quizz: marginal probability of event that we freeze when said probability exceeds q?

Freezing variables

Idea: progressively sample, stop when marginal probability becomes too high [MR98].



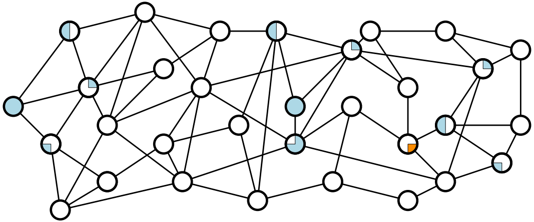
Fix threshold $q < p$. If marginal probability of an event reaches q, stop sampling its variables.

Quizz: marginal probability of event that we freeze when said probability exceeds q?

Could be as high as 1: for example, let $p = 1/k$, consider k biased coins of prob. $1/k^2$, let bad event = any of them is heads. When sampling, marginal probability can shot up to 1. [PT09]

Freezing variables

Idea: progressively sample, stop when marginal probability becomes too high [MR98].



Fix threshold $q < p$. If marginal probability of an event reaches q, stop sampling its variables.

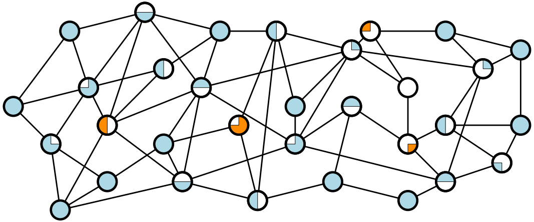
Quizz: marginal probability of event that we freeze when said probability exceeds q?

Could be as high as 1: for example, let $p = 1/k$, consider k biased coins of prob. $1/k^2$, let bad event = any of them is heads. When sampling, marginal probability can shot up to 1. [PT09]

To avoid this: replace LLL by approximate one using only fair coins. Each coinflip can only increase a probability by $\times 2$.

Freezing variables

Idea: progressively sample, stop when marginal probability becomes too high [MR98].



Fix threshold $q < p$. If marginal probability of an event reaches q, stop sampling its variables.

Quizz: marginal probability of event that we freeze when said probability exceeds q?

Could be as high as 1: for example, let $p = 1/k$, consider k biased coins of prob. $1/k^2$, let bad event = any of them is heads. When sampling, marginal probability can shot up to 1. [PT09]

To avoid this: replace LLL by approximate one using only fair coins. Each coinflip can only increase a probability by $\times 2$.

Fischer Ghaffari : distributed version

- Distance-2 color the graph
- Use it as a schedule for sampling

Reason for the d^2 in the runtime.

Fischer Ghaffari : distributed version

- Distance-2 color the graph
- Use it as a schedule for sampling

Reason for the d^2 in the runtime.

Why distance-2 ?

Fischer Ghaffari : distributed version

- Distance-2 color the graph
- Use it as a schedule for sampling

Reason for the d^2 in the runtime.

Why distance-2 ? an event never has two neighbors sampling some its variables being sampled at the same time.

This allows to do progressive sampling in $O(1)$ rounds even with many variables in an event.

In what sense should it shatter?

Intuitively, the probability that a node still has some unsampled variables with probability $\leq \frac{2p(d+1)}{q}$ at the end of the process, with freezing threshold $q/2$.

Why?

In what sense should it shatter?

Intuitively, the probability that a node still has some unsampled variables with probability $\leq \frac{2p(d+1)}{q}$ at the end of the process, with freezing threshold $q/2$.

Why? Consider a partial sampling of an event which stops due to the marginal probability reaching q, followed by sampling remaining variables. The probability for the event to hold is $\leq p$, but it's also $\geq q$ if starting from the partial sampling. So this stop in the process should have probability $\leq p/q$.

In what sense should it shatter?

Intuitively, the probability that a node still has some unsampled variables with probability $\leq \frac{2p(d+1)}{q}$ at the end of the process, with freezing threshold $q/2$.

Why? Consider a partial sampling of an event which stops due to the marginal probability reaching q, followed by sampling remaining variables. The probability for the event to hold is $\leq p$, but it's also $\geq q$ if starting from the partial sampling. So this stop in the process should have probability $\leq p/q$.

Assume $p = d^{-10}$. Set $q < \frac{1}{(d+1)}$ so bad events with unsampled variables still form a valid LLL ($eq(d+1) < 1$)
The probability of having unsampled variables is a small $1/\operatorname{poly}(d)$.

In what sense should it shatter?

Intuitively, the probability that a node still has some unsampled variables with probability $\leq \frac{2p(d+1)}{q}$ at the end of the process, with freezing threshold $q/2$.

Why? Consider a partial sampling of an event which stops due to the marginal probability reaching q, followed by sampling remaining variables. The probability for the event to hold is $\leq p$, but it's also $\geq q$ if starting from the partial sampling. So this stop in the process should have probability $\leq p/q$.

Assume $p = d^{-10}$. Set $q < \frac{1}{(d+1)}$ so bad events with unsampled variables still form a valid LLL $(eq(d+1) < 1)$
The probability of having unsampled variables is a small $1/\operatorname{poly}(d)$.

All we need is independence now...

Independence does not hold

What is independent: if we sample the variables of two non-adjacent events, whether the events holds is independent.

Independence does not hold

What is independent: if we sample the variables of two non-adjacent events, whether the events holds is independent.

What is NOT independent: when progressively sampling variables according to some rules in a process that takes $\Theta(d^2)$ rounds, whether two $O(d^2)$-distant events have unsampled variables at the end.

Independence does not hold

What is independent: if we sample the variables of two non-adjacent events, whether the events holds is independent.

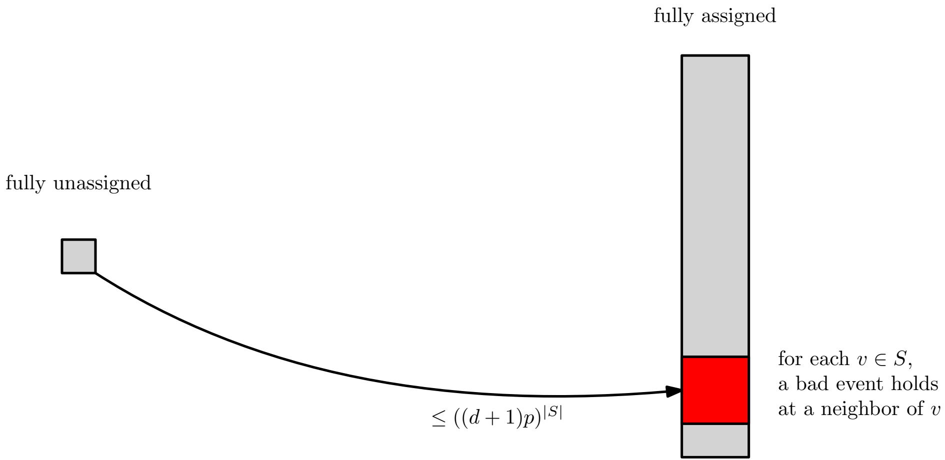
What is NOT independent: when progressively sampling variables according to some rules in a process that takes $\Theta(d^2)$ rounds, whether two $O(d^2)$-distant events have unsampled variables at the end.

We have a (tedious) concrete example.

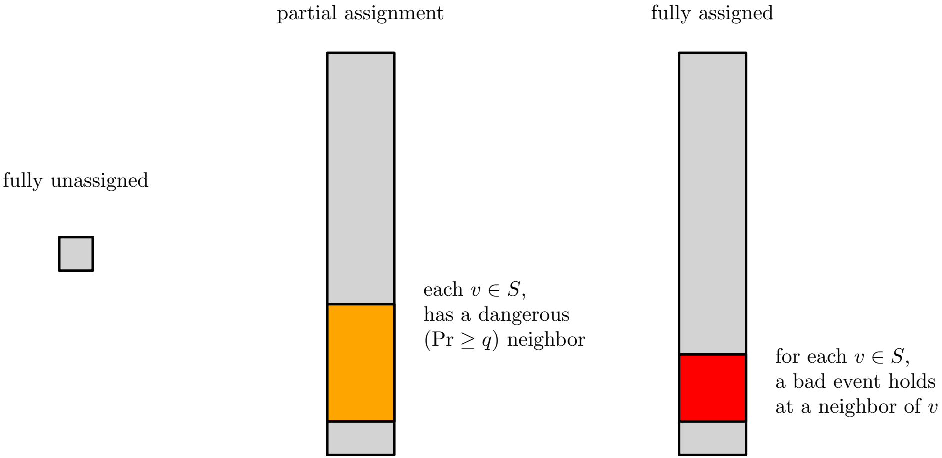
Visually


fully assigned
fully unassigned
for each $v \in S$,
a bad event holds
at a neighbor of v

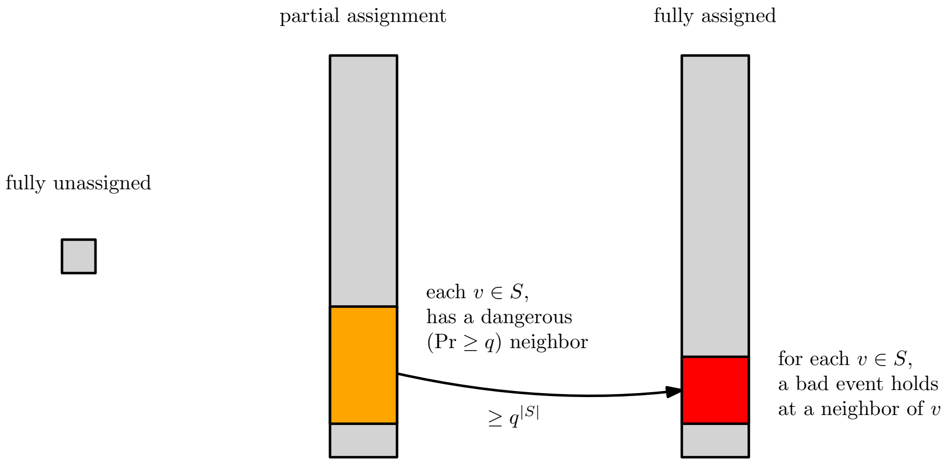
Visually


fully assigned
fully unassigned
for each $v \in S$,
a bad event holds
at a neighbor of v
$\leq ((d+1)p)^{|S|}$

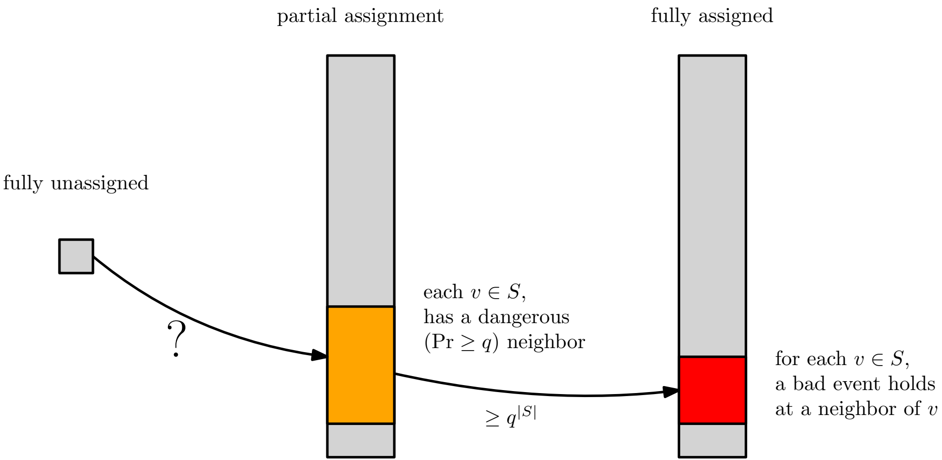
Visually


partial assignment
fully assigned
fully unassigned
each $v \in S$,
has a dangerous
($\Pr \geq q$) neighbor
for each $v \in S$,
a bad event holds
at a neighbor of v

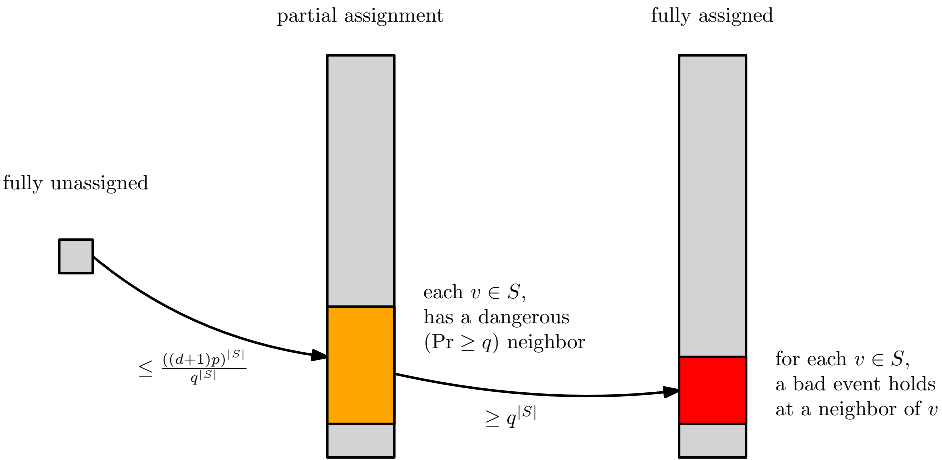
Visually


partial assignment
fully assigned
fully unassigned
each $v \in S$,
has a dangerous
(Pr $\geq q$) neighbor
$\geq q^{|S|}$
for each $v \in S$,
a bad event holds
at a neighbor of v

Visually


partial assignment
fully assigned
fully unassigned
?
each $v \in S$,
has a dangerous
(Pr $\geq q$) neighbor
$\geq q^{|S|}$
for each $v \in S$,
a bad event holds
at a neighbor of v

Visually


partial assignment
fully assigned
fully unassigned
each $v \in S$,
has a dangerous
$(\Pr \geq q)$ neighbor
$\leq \frac{((d+1)p)^{|S|}}{q^{|S|}}$
$\geq q^{|S|}$
for each $v \in S$,
a bad event holds
at a neighbor of v

Whitebox fix

1. Consider a set S of 4-distant nodes. Consider the event that they all have unsampled variables at the end of the partial sampling.

Whitebox fix

1. Consider a set S of 4-distant nodes. Consider the event that they all have unsampled variables at the end of the partial sampling.

2. Probability that starting from the partial sampling, we have a bad event next to each node in S? At least $q^{|S|}$.

Whitebox fix

1. Consider a set S of 4-distant nodes. Consider the event that they all have unsampled variables at the end of the partial sampling.

2. Probability that starting from the partial sampling, we have a bad event next to each node in S? At least $q^{|S|}$.

3. But if sampling everything at once initially, probability to have a bad event next to each node in S? At most $((d+1)p)^{|S|}$.

Whitebox fix

1. Consider a set S of 4-distant nodes. Consider the event that they all have unsampled variables at the end of the partial sampling.
2. Probability that starting from the partial sampling, we have a bad event next to each node in S? At least $q^{|S|}$.
3. But if sampling everything at once initially, probability to have a bad event next to each node in S? At most $((d+1)p)^{|S|}$.
4. **Conclusion:** probability to reach a partial assignment where all nodes in S have unsampled variables is at most

$$\frac{((d+1)p)^{|S|}}{q^{|S|}}\ .$$

pick $q = 1/(d+1)$, $p = d^{-10}/2$... done

Fischer-Ghaffari restored

Theorem ([FG17])

There is an algorithm of complexity $O(d^2 + T^{\mathsf{det}}_{\mathrm{LLL}}(\mathrm{poly}\log n))$ *for the constructive Lovász Local Lemma under criterion* $8pd^9 < 1$ *in the randomized LOCAL model.*

Fischer-Ghaffari restored

Theorem ([FG17])

There is an algorithm of complexity $O(d^2 + T^{\text{det}}_{\mathrm{LLL}}(\operatorname{poly}\log n))$ for the constructive Lovász Local Lemma under criterion $8pd^9 < 1$ in the randomized LOCAL model.

Last question, why $T^{\text{det}}_{\mathrm{LLL}}(\operatorname{poly}\log n))$ and not $T^{\text{det}}_{\mathrm{LLL}}(\operatorname{poly}(\Delta)\log n))$?

Fischer-Ghaffari restored

Theorem ([FG17])

There is an algorithm of complexity $O(d^2 + T^{\text{det}}_{\mathrm{LLL}}(\operatorname{poly}\log n))$ *for the constructive Lovász Local Lemma under criterion* $8pd^9 < 1$ *in the randomized LOCAL model.*

Last question, why $T^{\text{det}}_{\mathrm{LLL}}(\operatorname{poly}\log n))$ and not $T^{\text{det}}_{\mathrm{LLL}}(\operatorname{poly}(\Delta)\log n))$?

If $d^2 \geq \operatorname{poly}\log n$: use Moser-Tardos [MT10], faster than $O(d^2)$ in that case. Only shatter for $\Delta \in O(\operatorname{poly}\log n)$.

Opportunities

Opportunities

Many processes have the form: over the course of the algorithm, each node has at least x points in time where it has a good chance of making progress towards getting solved.

Opportunities

Many processes have the form: over the course of the algorithm, each node has at least x points in time where it has a good chance of making progress towards getting solved.

Making progress can be: simply getting solved, or, e.g., having its degree decrease by a constant factor.

Opportunities

Many processes have the form: over the course of the algorithm, each node has at least x points in time where it has a good chance of making progress towards getting solved.

Making progress can be: simply getting solved, or, e.g., having its degree decrease by a constant factor.

We give a generic argument that captures many algorithms for greedy problems (MIS, MM, coloring...)

More concretely

If each node in a set has x guaranteed steps where it gets solved w.p. $\geq 1 - p$, conditioned on an arbitrary past.

Probability that no one in the set gets solved: $\leq p^{|S|}$.

Examples: $O(\log \Delta)$ shattering by Ghaffari for MIS, [Gha16], coloring and MIS by Barenboim, Elkin, Pettie and Schneider [BEPS16].

Conclusion

Do all results in the literature using shattering hold? Fortunately, many seem to do.

Conclusion

Do all results in the literature using shattering hold? Fortunately, many seem to do.

> *Human-generated proofs are meaningful and (almost always) directionally correct — David Bessis*

Conclusion

Do all results in the literature using shattering hold? Fortunately, many seem to do.

> *Human-generated proofs are meaningful and (almost always) directionally correct — David Bessis*

Is there a more generic argument for LLL?

Conclusion

Do all results in the literature using shattering hold? Fortunately, many seem to do.

> *Human-generated proofs are meaningful and (almost always) directionally correct — David Bessis*

Is there a more generic argument for LLL?

Thanks!

Small announcement: PODC 2027 in Iceland

June 22 to 24, 2027.
Reykjavik University

Leonid Barenboim, Michael Elkin, Seth Pettie, and Johannes Schneider.
The locality of distributed symmetry breaking.
J. ACM, 63(3):20:1–20:45, 2016.

Yi-Jun Chang and Seth Pettie.
A time hierarchy theorem for the LOCAL model.
SIAM J. Comput., 48(1):33–69, 2019.

Kai-Min Chung, Seth Pettie, and Hsin-Hao Su.
Distributed algorithms for the Lovász local lemma and graph coloring.
Distributed Comput., 30(4):261–280, 2017.

Manuela Fischer and Mohsen Ghaffari.
Sublogarithmic distributed algorithms for lovász local lemma, and the complexity hierarchy.
In *Proceedings of the International Symposium on Distributed Computing (DISC)*, LIPIcs, pages 18:1–18:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

Mohsen Ghaffari.
An improved distributed algorithm for maximal independent set.
In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 270–277. SIAM, 2016.

Mohsen Ghaffari, David G. Harris, and Fabian Kuhn.
On derandomizing local distributed algorithms.
In *Proceedings of the Symposium on Foundations of Computer Science (FOCS)*, pages 662–673. IEEE Computer Society, 2018.

Mohsen Ghaffari and Hsin-Hao Su.
Distributed degree splitting, edge coloring, and orientations.
In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2505–2523. SIAM, 2017.

Michael Molloy and Bruce A. Reed.
Further algorithmic aspects of the local lemma.

In *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, pages 524–529. ACM, 1998.

Robin A. Moser and Gábor Tardos.
A constructive proof of the general Lovász local lemma.
J. ACM, 57(2):11:1–11:15, 2010.

János Pach and Gábor Tardos.
Conflict-free colourings of graphs and hypergraphs.
Comb. Probab. Comput., 18(5):819–834, 2009.