New Constructive Aspects of the Lovász Local Lemma, and their Applications

Aravind Srinivasan
University of Maryland, College Park

June 15, 2011

Collaborators: Bernhard Haeupler (MIT) & Barna Saha (UMD)

Algorithmic versions of the LLL

$\mathcal{A} = \{A_1, A_2, \ldots, A_m\}$: "bad" events, each defined by indep. random variables $X_1, X_2, \ldots, X_n$.

Ubiquitous version with neigborhood relation Γ on $\mathcal{A}$.

Are all A_i *simultaneously* avoidable?
Output = assignment to all X_j; output size = n.

Algorithmic versions of the LLL

$\mathcal{A} = \{A_1, A_2, \ldots, A_m\}$: "bad" events, each defined by <u>indep</u>. random variables $X_1, X_2, \ldots, X_n$.

Ubiquitous version with neigborhood relation Γ on $\mathcal{A}$.

Are all A_i *simultaneously* avoidable?
Output = assignment to all X_j; output size = n.

Main results:

- "Any" LLL application $\to$ poly(n)-time alg. (even if $m \gg$ poly(n)), if we give a tiny slack in the LLL-condition;
- MAX SAT–like problems: avoiding "most" A_i (algorithmically) – interpolation between linearity of expectation and LLL.

LLL: symmetric version

"$\Pr[\text{no } A_i] > 0$": Union Bound $\sum_i \Pr[A_i] < 1$ often too weak.

LLL (symmetric version): Suppose

- $\max_i \Pr[A_i] \leq p$, and
- each A_i has $\leq D$ neighbors.

Then, $e \cdot p \cdot (D+1) \leq 1$ implies $\Pr[\text{no } A_i \text{ holds}] > 0$.

Numerous applications. Typical case: $D \ll m$.

LLL: symmetric version

"$\Pr[\text{no } A_i] > 0$": Union Bound $\sum_i \Pr[A_i] < 1$ often too weak.

LLL (symmetric version): Suppose

- $\max_i \Pr[A_i] \leq p$, and
- each A_i has $\leq D$ neighbors.

Then, $e \cdot p \cdot (D+1) \leq 1$ implies $\Pr[\text{no } A_i \text{ holds}] > 0$.

Numerous applications. Typical case: $D \ll m$.

Algorithmic version?
$\Pr[\bigwedge_i \overline{A_i}]$ inevitably small:

- Choose indep. set I of the A_i with $|I| \geq m/(D+1)$.
- $\Pr[\bigwedge_i \overline{A_i}] \leq \Pr[\bigwedge_{i \in I} \overline{A_i}] = (1-p)^{m/(D+1)} \approx \exp(-mp/D)$.

Application: Domatic Partitions

Feige-Halldórsson-Kortsarz-S: a maximization problem with a logarithmic apx. threshold.

Graph G; $N^+(v) =$ inclusive neighborhood of vertex v.

Partition vertices into a max. # dominating sets: i.e., "color" vertices with max. # colors so that

$$\forall \text{ vertices } v, \text{ all colors visible in } N^+(v).$$

[Chen-Jamieson-Balakrishnan-Morris]: wireless coordination.
If $(\delta, \Delta) =$ (min., max.) degrees, $\text{OPT} \leq \delta + 1$.

Application: Domatic Partitions

Feige-Halldórsson-Kortsarz-S: a maximization problem with a logarithmic apx. threshold.

Graph G; $N^+(v)$ = inclusive neighborhood of vertex v.

Partition vertices into a max. # dominating sets: i.e., "color" vertices with max. # colors so that

$$\forall \text{ vertices } v, \text{ all colors visible in } N^+(v).$$

[Chen-Jamieson-Balakrishnan-Morris]: wireless coordination.
If $(\delta, \Delta) =$ (min., max.) degrees, $\text{OPT} \leq \delta + 1$.

[FHKS]: apx. threshold $= \ln \Delta$. Here: $3 \ln d$–apx. for d-regular G.

Domatic partitions assuming d-regularity

Randomly color vertices using $\ell \sim d/(3 \ln d)$ colors.
Bad event $A_{v,c}$: "c not visible at v".

$p = \Pr[A_{v,c}] = (1 - 1/\ell)^{d+1} \sim 1/d^3$.

Dependence of fixed $A_{v,c}$?
Only on $A_{w,c'}$ with $\mathrm{dist}(v, w) \leq 2$.

$\#w < d^2$; $\#c' \leq \ell$. So, $D < d^3/(3 \ln d)$.

$e \cdot p \cdot (D + 1) \leq 1$; thus $\exists$ good coloring.

Domatic partitions assuming d-regularity

Randomly color vertices using $\ell \sim d/(3 \ln d)$ colors.
Bad event $A_{v,c}$: "c not visible at v".

$p = \Pr[A_{v,c}] = (1 - 1/\ell)^{d+1} \sim 1/d^3$.

Dependence of fixed $A_{v,c}$?
Only on $A_{w,c'}$ with $\text{dist}(v, w) \leq 2$.

$\#w < d^2$; $\#c' \leq \ell$. So, $D < d^3/(3 \ln d)$.

$e \cdot p \cdot (D + 1) \leq 1$; thus $\exists$ good coloring.

Correct constant "3" $\rightarrow$ "1": iterated app. of LLL, a powerful methodology ([Molloy-Reed]: "Graph colouring and the probabilistic method").

LLL: Asymmetric Version

LLL, general "asymmetric" version: If $\exists x : \mathcal{A} \rightarrow (0,1)$ such that

$$\forall i : \Pr[A_i] \le x(A_i) \prod_{A_j \in \Gamma(A_i)} (1 - x(A_j)),$$

then $\Pr[\bigwedge_i \overline{A_i}] \ge \prod_i (1 - x(A_i)) > 0$.

Numerous applications:

- (Hyper-)Graph Colorings and Ramsey Numbers
- Routing [Leighton-Maggs-Rao]
- LP-Integrality gaps [Feige, Leighton-Lu-Rao-S]
- Edge-disjoint paths [Andrews] ...

The Trivial Algorithm

The Trivial Algorithm:

repeat
  pick a random assignment for all X_j
until no A_i holds

The Trivial Algorithm

The Trivial Algorithm:

repeat
  pick a random assignment for all X_j
until no A_i holds

Theorem (LLL)

If the LLL-conditions hold, then the above algorithm finds a satisfying assignment with positive probability.

The Trivial Algorithm:

repeat
  pick a random assignment for all X_j
until no A_i holds

Theorem (LLL)

If the LLL-conditions hold, then the above algorithm finds a satisfying assignment with positive probability.

BUT: Run-time usually exponential in m (let alone n).

The Moser-Tardos Breakthrough

Algorithmic versions of the LLL: [Beck, Alon, Molloy-Reed, Czumaj-Scheideler, S, Moser, ...] culminating in MT:

The MT Algorithm:

start with an arbitrary assignment
while $\exists$ event A_i that holds **do**
  assign new random values to the variables of A_i

Algorithmic versions of the LLL: [Beck, Alon, Molloy-Reed, Czumaj-Scheideler, S, Moser, ...] culminating in MT:

The MT Algorithm:

start with an arbitrary assignment
while $\exists$ event A_i that holds **do**
  assign new random values to the variables of A_i

Theorem (MT)

If the LLL-conditions hold, then the above algorithm finds a satisfying assignment within an expected $\sum_i \frac{x(A_i)}{1-x(A_i)}$ *iterations.*

LLL-distribution and the MT-Algorithm

The trivial algorithm outputs a random sample from the *conditional LLL-distribution* $\mathcal{D}$, the distribution that conditions on avoiding all A_i.

LLL-distribution and the MT-Algorithm

The trivial algorithm outputs a random sample from the *conditional LLL-distribution* $\mathcal{D}$, the distribution that conditions on avoiding all A_i.

A Well-known Bound

For any event $B = B(X_1, X_2, \ldots, X_n)$,

$$\Pr_{\mathcal{D}}(B) := \Pr\left(B \mid \bigwedge_i \overline{A_i}\right) \le \Pr(B) \cdot \left(\prod_{A_j \in \Gamma(B)} (1 - x(A_j))\right)^{-1}$$

(1)

LLL-distribution and the MT-Algorithm

The trivial algorithm outputs a random sample from the *conditional LLL-distribution* $\mathcal{D}$, the distribution that conditions on avoiding all A_i.

A Well-known Bound

For any event $B = B(X_1, X_2, \ldots, X_n)$,

$$\Pr_{\mathcal{D}}(B) := \Pr\left(B \mid \bigwedge_i \overline{A_i}\right) \leq \Pr(B) \cdot \left(\prod_{A_j \in \Gamma(B)} (1 - x(A_j))\right)^{-1}$$

(1)

Theorem

The output distribution of the MT-algorithm satisfies (1).

LLL-Applications with Exponentially Many Events

Examples:

- Acyclic edge coloring
- Non-repetitive coloring
- Santa Claus problem
- Edge-disjoint paths, . . .

Examples:

- Acyclic edge coloring
- Non-repetitive coloring
- Santa Claus problem
- Edge-disjoint paths, ...

Problems with running MT:

1. E[# resamplings]: $\sum_i \frac{x(A_i)}{1-x(A_i)}$

LLL-Applications with Exponentially Many Events

Examples:

- Acyclic edge coloring
- Non-repetitive coloring
- Santa Claus problem
- Edge-disjoint paths, ...

Problems with running MT:

1. E[# resamplings]: $\sum_i \frac{x(A_i)}{1-x(A_i)}$
2. representation of the bad events

Examples:

- Acyclic edge coloring
- Non-repetitive coloring
- Santa Claus problem
- Edge-disjoint paths, . . .

Problems with running MT:

1. E[# resamplings]: $\sum_i \frac{x(A_i)}{1-x(A_i)}$
2. representation of the bad events
3. verifying a solution / finding some A_i that holds currently

Solving Problem 1

Theorem

Let $\delta = \min_i \Pr[A_i]$. *Then,*

$$\begin{aligned} E[\#\ \textit{iterations of MT}] &\leq \sum_i \frac{x(A_i)}{1 - x(A_i)} \\ &\leq (\sum_i x(A_i)) \cdot \max_i \frac{1}{1 - x(A_i)} \\ &\leq O(n \log(1/\delta)) \cdot \max_i \frac{1}{1 - x(A_i)}. \end{aligned}$$

In all app.s known to us, $\log \frac{1}{\delta} \leq O(n \log n)$.

Solving Problem 2+3

How do we represent the events (implicitly) s.t.

- checking a solution or
- finding some A_i that holds currently

can be done in poly(n) time?

Hopeless:
In most applications this is (NP-)hard

Solving Problem 2+3

How do we represent the events (implicitly) s.t.

- checking a solution or
- finding some A_i that holds currently

can be done in poly(n) time?

Hopeless:
In most applications this is (NP-)hard

Algorithm: Run MT on a core-subset of the $\mathcal{A}$, of poly(n) size.

Solving Problem 2+3

How do we represent the events (implicitly) s.t.

- checking a solution or
- finding some A_i that holds currently

can be done in poly(n) time?

Hopeless:
In most applications this is (NP-)hard

Algorithm: Run MT on a core-subset of the $\mathcal{A}$, of poly(n) size.

Analysis:

- Bound the probabilities of non-core events using (1)

Solving Problem 2+3

How do we represent the events (implicitly) s.t.

- checking a solution or
- finding some A_i that holds currently

can be done in poly(n) time?

Hopeless:
In most applications this is (NP-)hard

Algorithm: Run MT on a core-subset of the $\mathcal{A}$, of poly(n) size.

Analysis:

- Bound the probabilities of non-core events using (1)
- Use a union bound over these probabilities to prove that with high probability all the A_i are avoided.

Solving Problem 2+3

Theorem

If $\exists\epsilon \in (0,1)$ *such that for all* A_i,

$$\Pr[A]^{1-\epsilon} \leq x(A_i) \cdot \prod_{A_j \in \Gamma(A_i)} (1 - x(A_j)),$$

then:

- *for any* $p \geq \frac{1}{poly(n)}$, $|\{A_i : \Pr[A_i] \geq p\}| \leq poly(n)$;
- *If* $\log\frac{1}{\delta} \leq poly(n)$ *and the above core is "checkable", then for any desired constant* $c > 0$, $\exists$ *Monte Carlo alg. (with* $p \sim n^{-c/\epsilon}$*) that terminates within* $O(\frac{n}{\epsilon}\log\frac{n}{\epsilon})$ *resamplings and returns a good assignment with probability at least* $1 - n^{-c}$.

Algorithmic Results

The first/best- known efficient algorithms for:

- $O(1)$–apx. for the Santa Claus problem (Feige's proof made constructive)
- non-repetitive coloring (proof of Alon-Grytczuk-Hauszczak-Riordan made constructive)
- acyclic edge coloring
- edge-disjoint paths (Andrews [2010])

Allowing some A_i to hold

Interpolating between the LLL and linearity of expectation:

Theorem

In the symmetric LLL with p and D, if $D \le \alpha \cdot (1/(ep) - 1)$ $(1 < \alpha < e)$ *then we can make at most* $\sim (e \ln(\alpha)/\alpha) \cdot mp$ *of the* A_i *to hold, in randomized poly(m) time.*

Open Problems

- Is "$e \ln(\alpha)/\alpha$" tight?
- Derandomization
- Further analysis of dependencies among *non-core* events
- How much slack is really needed?
- Lopsided Local Lemma?
- Full understanding of [Kolipaka-Szegedy] setting

Thank you!

Questions?