This is a summary of the talk I gave on December 7th of 2010 in University of Waterloo. The aim is to describe the main contributions and ideas of the paper

> New Constructive Aspects of the Lovasz Local Lemma, by Bernhard Haeupler, Barna Saha, Aravind Srinivasan (FOCS'10).

Abbas Mehrabian

The Setup

$\mathcal{P}$ is a finite set of n independent random variables.

$\mathcal{A}$ is a family of m ("bad") events that are determined by these variables.

$\mathrm{vbl}(A)$ is the minimal set of variables that determine A.

Violation: An assignment of variables violates event A if it makes A happen.

Good assignment: An assignment is good if it does not violate any event.

$\Gamma(B) = \Gamma_{\mathcal{A}}(B)$ is the set of events $A \in \mathcal{A}$ such that $A \neq B$ and $\mathrm{vbl}(A), \mathrm{vbl}(B)$ intersect.

LLL conditions: $\mathcal{A}$ satisfies LLL conditions if there exists a function $x : \mathcal{A} \to (0,1)$ such that for all events A,

$$\mathbf{Pr}(A) \leq x(A) \prod_{B \in \Gamma(A)} (1 - x(B)) \, .$$

Dependency graph: A simple graph $G = G_{\mathcal{A}}$ that has vertex set $\mathcal{A}$ with A, B adjacent if $A \neq B$ and $\mathrm{vbl}(A), \mathrm{vbl}(B)$ intersect.

δ is $\min_A x(A) \prod_{B \in \Gamma(A)} (1 - x(B))$.

Efficient verifiability: A set of events is called efficiently verifiable if, given an assignment for $\mathcal{P}$, one can efficiently find a violated event, or detect that there is no such event.

$\Delta = \Delta(H)$ is the maximum degree of graph H.

A Review of Lovász Local Lemma
Lovász Local Lemma (Erdös, Lovász'75). *If $\mathcal{A}$ satisfies LLL conditions then there exists a good assignment.*
Example. Consider a k-CNF formula such that each clause shares variables with at most $2^k/e - 1$ other clauses. Then there exists an assignment to the variables such that all clauses are satisfied.

Proof. Give each variable a random (0 or 1) value with equal probability. Every clause corresponds to a bad event whose probability is 2^{-k}. Let $x(A) := e/2^k$ for all such bad events. Then we have

$$\frac{1}{2^k} \leq \frac{e}{2^k}(1 - \frac{e}{2^k})^{\frac{2^k}{e}-1},$$

where we have used the inequality

$$(1 - \frac{1}{d})^{d-1} \geq \frac{1}{e}.$$

Hence LLL conditions are satisfied and there exists a good assignment by Lovász Local Lemma. $\square$

The Moser-Tardos Algorithm

```
start with a random assignment for P
while there exists a violated event
   re-sample the variables of an arbitrary violated event A
```

Theorem 1 (Moser, Tardos'09)**.** *If LLL conditions are satisfied, then this algorithm finds a good assignment and the expected number of re-samplings is at most*

$$\sum_A \frac{x(A)}{1-x(A)}$$

The size of the problem is usually poly(n), so if $m = |\mathcal{A}|$ is super-polynomial in n, then the bound on the expected running time is not polynomial. However,

Theorem 2 (3.1)**.** *We have $\sum_{A\in\mathcal{A}} x(A) \leq n\log(1/\delta)$ so the expected number of re-samplings of the MT-algorithm is at most*

$$\left(\sum_{A\in\mathcal{A}} x(A)\right)\left(\max_{A\in\mathcal{A}} \frac{1}{1-x(A)}\right) \leq \left(n\log\frac{1}{\delta}\right)\left(\max_{A\in\mathcal{A}} \frac{1}{1-x(A)}\right).$$

Remarks:

- $\log(1/\delta) = O(n\log n)$ in all known applications. In fact $\min_{A\in\mathcal{A}} \mathbf{Pr}(A) \leq \delta$.
- In all known applications, all $x(A)$'s are at least $1/2$ and so $\max_{A\in\mathcal{A}} \frac{1}{1-x(A)} \leq 2$.
- The $\sum_{A\in\mathcal{A}} x(A) \leq n\log\frac{1}{\delta}$ bound is tight asymptotically, but the δ is usually not that small.

Although this is a good bound on running time (polynomial in all known applications), still if there are super-polynomially many events, finding a violated event might require super-polynomial time. If the set of bad events is efficiently verifiable, however, we find a polynomial algorithm.

Acyclic Edge Colouring

Definition. A colouring of the edges of a graph is acyclic, if incident edges get different colours, and among the edges of every cycle at least three different colours appear. Let $a(H)$ be the minimum number of colours needed for an acyclic edge colouring of a graph H.

Theorem 3 (Molloy, Reed'98)**.** $a(H) \le 16\Delta(H)$.

Proof. Set $C := 16\Delta$ and colour each edge of H randomly using one of the colours $1, 2, \ldots, C$. A type 1 (bad) event corresponds to two incident edges getting the same colour, and a type k (bad) event corresponds to a cycle of length $2k$ to get at most two colours. Each type k event depends on at most $4k\Delta$ type 1 events, and at most $2k\Delta^{2(l-1)}$ type l events for every $l > 1$. Set $x(A) := (2/C)^{2(k-1)}$ for all events A of type $k \ge 1$. Then the probability of an event of type k is at most

$$\begin{aligned}
2\frac{C}{2}^{-2(k-1)} &\le (\frac{2}{C})^{2(k-1)} e^{-k} \prod_{l>1}\left(e^{-4(1/8)^{2(l-1)}}\right) \\
&= (\frac{2}{C})^{2(k-1)} (e^{-\frac{4}{C}})^{4k\Delta} \prod_{l>1}\left(e^{-2(\frac{2}{C})^{2(l-1)}}\right)^{2k\Delta^{2(l-1)}} \\
&\le (\frac{2}{C})^{2(k-1)} (1-\frac{2}{C})^{4k\Delta} \prod_{l>1}\left(1-(\frac{2}{C})^{2(l-1)}\right)^{2k\Delta^{2(l-1)}} \\
&= x_k(1-x_1)^{4k\Delta}\prod_{l>1}(1-x_l)^{2k\Delta^{2(l-1)}}
\end{aligned}$$

and so LLL conditions are satisfied. Therefore by Lovász Local Lemma, an acyclic edge colouring exists for H that uses $16\Delta(H)$ colours. □

Theorem 4 (7.1)**.** *There is a randomized algorithm that produces an acyclic edge colouring of any graph in expected polynomial time using* 16Δ *colours.*

Proof. We will use Theorem 2. We have $\delta \ge \min \mathbf{Pr}(A) \ge C^{-2n}$ so $\log(1/\delta) = O(n \log \Delta)$. Also the set of bad events is efficiently verifiable: Violated type 1 bad events are easy to detect, and to find a violated event of type $k > 1$, consider the subgraph induced on every pair of colours and check if it has a cycle. □

For graphs with girth $\Omega(\Delta \log \Delta)$ we have $a(H) \le \Delta(H) + 2$ and an acyclic edge colouring can be found using the same idea in expected polynomial time. It has been conjectured that $a(H) \le \Delta(H) + 2$ holds for all graphs H.

The MT-algorithm's Output's Distribution

Theorem 5 (2.2)**.** *Assume that LLL conditions hold. For any event B that is determined by $\mathcal{P}$, the probability that the output of MT-algorithm violates B is at most*

$$\mathbf{Pr}(B)\prod_{C\in\Gamma_{\mathcal{A}}(B)}(1-x(C))^{-1}$$

Remark. The same upper bound holds for $\mathbf{Pr}(B|\text{no } A\in\mathcal{A} \text{ happen})$.

Theorem 6 (3.4)**.** *Assume that $\log\frac{1}{\delta}\leq n^a$ for some fixed a. Suppose that there is a constant $\epsilon\in(0,1)$ and a function $x:\mathcal{A}\to(0,1-\epsilon)$ such that for all $A\in\mathcal{A}$,*

$$\mathbf{Pr}(A)^{1-\epsilon}\leq x(A)\prod_{B\in\Gamma_{\mathcal{A}}(A)}(1-x(B)).$$

Then for any c there exists an expected polynomial time Monte Carlo algorithm that returns a good assignment with probability at least $1-n^{-c}$.

Proof. For all events A,

$$\mathbf{Pr}(A)\leq\mathbf{Pr}(A)^{1-\epsilon}\leq x(A)<1-\epsilon$$

so $\mathbf{Pr}(A)^{\epsilon}\leq 1$ and in fact $\mathbf{Pr}(A)\leq x(A)\prod_{B\in\Gamma(A)}(1-x(B))$. That is, $\mathcal{A}$ satisfies LLL conditions.

Let $p:=n^{\frac{c-a-1}{\epsilon}}$ and $\mathcal{A}':=\{A:\mathbf{Pr}(A)\geq p\}$. By Theorem 2, $\sum x(A)\leq n\log(1/\delta)$ so $|\mathcal{A}'|\leq n\log(1/\delta)/p=\text{poly}(n)$. Thus $\mathcal{A}'$ is efficiently verifiable and satisfies LLL conditions, hence a good assignment for $\mathcal{A}'$ can be found in expected polynomial time using the MT-algorithm.

For an output of the MT-algorithm, the probability of violating an event in $\mathcal{A}-\mathcal{A}'$ is bounded from above by

$$\begin{aligned}\sum_{B\in\mathcal{A}-\mathcal{A}'}\mathbf{Pr}(B)\prod_{C\in\Gamma_{\mathcal{A}'}(B)}(1-x(C))^{-1}\leq&\sum_{B\in\mathcal{A}-\mathcal{A}'}p^{\epsilon}\mathbf{Pr}(B)^{1-\epsilon}\prod_{C\in\Gamma_{\mathcal{A}'}(B)}(1-x(C))^{-1}\\ \leq& p^{\epsilon}\sum_{B\in\mathcal{A}-\mathcal{A}'}x(B)\leq p^{\epsilon}n\log(\frac{1}{\delta})\leq n^{-c}\end{aligned}$$

using union bound and Theorem 6. $\square$

Non-repetitive Coloring of Graphs

Definition. Let H be a graph and $c : V(H) \to \{1, \dots, C\}$ be a colouring of its vertices. Colouring c is called non-repetitive if for every $k \geq 1$ and every path $v_1 v_2 \dots v_{2k}$ in H, there exists an index i with $c(v_i) \neq c(v_{i+k})$. That is, the sequence of the colours appearing in this path, is not of the form xx for some sequence x of length k. The minimum C for which such a colouring possible is denoted by $\pi(H)$.

Theorem 7 (Alon, Grytczuk, Haluszczak, Riordan'02)**.**

$$\pi(H) \leq C := \lceil 2e^{16} \Delta(H)^2 \rceil.$$

Proof. Colour the vertices independently and randomly using one of the colours $1, 2, \dots, C$. Every path of length $2i$ corresponds to a bad event, which we call an event of type i, whose probability is C^{-i}. A path of length $2i$ intersects at most $4ij\Delta^{2j}$ paths of length $2j$. For an event A of type i, set $x(A) := (2\Delta^2)^{-i}$. The probability of an event of type i is

$$(2e^{16}\Delta^2)^{-i} \leq \frac{1}{2^i \Delta^{2i}} \prod_{j>1} (e^{-8i})^{j/2^j} = x_i \prod_{j>1} e^{-2x_j \times 4ij\Delta^{2j}} \leq x_i \prod_{j>1} (1 - x_j)^{4ij\Delta^{2j}},$$

which shows that LLL conditions hold, thus there exists a non-repetitive colouring using C colours. $\square$

Theorem 8 (5.2)**.** *For any fixed $\epsilon \in (0, 1)$, the above argument can be turned into an expected polynomial algorithm, which produces a non-repetitive colouring using $C' := C^{\frac{1}{1-\epsilon}}$ colours with high probability.*

Proof. The conditions of Theorem 6 are satisfied as

$$C'^{-i(1-\epsilon)} = C^{-i} \leq x_i \prod_{j>1} (1 - x_j)^{4ij\Delta^{2j}},$$

and also

$$\delta \geq \min \mathbf{Pr}(A) \geq C'^{-n} = \Delta^{-\frac{2}{1-\epsilon} n}$$

shows that $\log(1/\delta) = O(n \log \Delta)$ and we are done (by putting e.g. c=1). $\square$

Two More Applications
The Santa Claus Problem:

- There are n items to be distributed among m persons.
- Each item j has value p_j.
- Each player either likes item j or does not like it, and its utility is the sum of the values of the liked obtained items.
- We want to maximize the minimum utility of the players.
- The problem is NP-hard and has no better than 1/2-approximation.
- An LP relaxation was considered and resulted in an $O(\log \log \log m / \log \log m)$-approximation algorithm (Bansal, Sviridenko'06).
- Two years later Feige proved using Lovász Local Lemma that the integrality gap is constant.
- Theorem 6 provides a randomized algorithm with a constant approximation ratio.

Constructive Lower Bounds For Ramsey Numbers:

- Some of the lower bounds for Ramsey numbers use Lovász Local Lemma.
- Theorem 6 provides constructive lower bounds for those Ramsey numbers.

Violating Few Bad Events - An Example

Theorem 9 (8.1)**.** *Assume that $\mathcal{F}$ is a k-CNF formula with n variables and m clauses, and there exists $\mathcal{F}' \subseteq \mathcal{F}$ such that*

(i) every clause in $\mathcal{F}'$ shares variables with at most $2^k/e - 1$ clauses in $\mathcal{F}'$, and

(ii) every clause in $\mathcal{F} - \mathcal{F}'$ shares variables with at most $\gamma(2^k/e - 1)$ many clauses in $\mathcal{F}'$, for some $\gamma \geq 0$.

Then for any $\theta = 1/poly(m, n)$, there exists a randomized $poly(n, m)$-time algorithm that produces, with high probability, an assignments in which all clauses in $\mathcal{F}'$ are satisfied, and at most a $(1 + \theta)2^{-k}e^{\gamma}$ fraction of clauses from $\mathcal{F} - \mathcal{F}'$ are unsatisfied.

Proof. Every clause corresponds to a bad event A. Setting $x(A) = e/2^k$ for all $A \in \mathcal{F}'$ shows that LLL conditions hold for $\mathcal{F}'$. Run the MT-algorithm on $\mathcal{F}'$ for n^c times its expected running time. It will finish with probability $1 - n^{-c}$, and by Theorem 5 the probability that a clause in $\mathcal{F} - \mathcal{F}'$ is not satisfied is at most

$$2^{-k}(1 - e/2^k)^{-\gamma(2^k/e-1)} \leq e^{\gamma}2^{-k}.$$

Thus the expected fraction of unsatisfied clauses is at most $e^{\gamma}2^{-k}$. The probability that more than $(1+\theta)2^{-k}e^{\gamma}$ fraction of clauses from $\mathcal{F}-\mathcal{F}'$ are unsatisfied is at most $(\frac{1}{1+\theta})$. Repeating this process for a suitably large number of times gives you a satisfying assignment with arbitrarily small failure probability. $\square$

Violating Few Bad Events - A General Theorem

Theorem 10 (8.3)**.** *Let $1 < \alpha < e$ be fixed. Assume that the probability of every $A \in \mathcal{A}$ is at most $p = o(1)$ and the maximum degree in the dependency graph $G_{\mathcal{A}}$ is at most $d = \alpha(1/(ep) - 1)$. Then there exists an assignment that violates at most $(1 + o(1))mpe \ln(\alpha)/\alpha$ events.*

Proof. Let $\mathcal{A}'$ be a random subset obtained by choosing each $A \in \mathcal{A}$ with probability $1 - \ln \alpha$. Delete those events that have more than $d(1 - \ln \alpha)$ neighbours in $\mathcal{A}'$. By Chernoff's bound, with high probability $|\mathcal{A}'| = (1 - o(1))m(1 - \ln \alpha)$, and just a $o(p)$ fraction of events has been deleted from $\mathcal{A}$. By putting $x(A) = 1/d$ for all $A \in \mathcal{A}'$ and verifying that

$$p \leq \frac{1}{d}(1 - \frac{1}{d})^{d(1-\ln \alpha)},$$

one sees that $\mathcal{A}'$ satisfies LLL conditions, so there exists an assignment with no violated $A \in \mathcal{A}'$.

The probability that an undeleted $B \in \mathcal{A} - \mathcal{A}'$ is violated is at most

$$p(1 - \frac{1}{d})^{-d(1-\ln \alpha)}$$

by Theorem 5. So the expected number of total violated events is at most

$$o(p) + ((1 + o(1))(\ln \alpha)m)\left(p(1 - \frac{1}{d})^{-d(1-\ln \alpha)}\right) \leq (1 + o(1))mpe \ln(\alpha)/\alpha$$

with high probability. Hence there exists an assignment with at most this many violated events. □

Remarks:

- If there is no bound on the maximum degree, then by linearity of expectation it can be checked that there exists an assignment violating at most mp events, and if $e > \alpha$ then there are examples in which it is not possible to violate less than mp events.

- If $\alpha \leq 1$ then Lovász Local Lemma can be used directly on $\mathcal{A}$ to show that there is an assignment with no violated event.

- If $\mathcal{A}$ is efficiently verifiable, then this proof can be made into an expected polynomial time algorithm by generating $\mathcal{A}'$ explicitly and executing the MT-algorithm on it. (the expected running-time of the MT-algorithm will be bounded by $O(n^2 \log d)$ by Theorem 2.) The expected number of violated events in the output would be the same.