An extension of the Moser-Tardos algorithmic local lemma

Wesley Pegden*

January 26, 2013

Abstract

A recent theorem of Bissacot, et al. proved using results about the cluster expansion in statistical mechanics extends the Lovász Local Lemma by weakening the conditions under which its conclusion holds. In this note, we prove an algorithmic analog of this result, extending Moser and Tardos's recent algorithmic Local Lemma, and providing an alternative proof of the theorem of Bissacot, et al. applicable in the Moser-Tardos algorithmic framework.

1 Introduction

If events $A_1, A_2, \ldots, A_n$ are independent, then we have $\mathrm{P}(\bigcap \bar{A}_i) > 0$ so long as $\mathrm{P}(A_i) < 1$ for each i. A central tool in probabilistic combinatorics is the Lovász Local Lemma proved by Erdős and Lovász [5], which can be seen as generalizing this simple fact to situations where some dependencies among the A_i are allowed, in exchange for good bounds on the probabilities $\mathrm{P}(A_i)$.

The Local Lemma is commonly presented through the framework of a *dependency graph* on the events A_i, where if $\mathcal{C}$ is any family of non-neighbors of some A_i, then we have that A_i is independent of the family $\mathcal{C}$ of events. The Lovász Local Lemma is then as follows:

Theorem 1.1 (Lovász Local Lemma)**.** *Let G be any dependency graph for a finite family $\mathcal{A}$ of events, and suppose that there are real numbers $0 < x_A < 1$ $(A \in \mathcal{A})$ such that for all $A \in \mathcal{A}$ we have*

$$\mathrm{P}(A) \leq x_A \prod_{B \sim A} (1 - x_B), \tag{1}$$

where $B \sim A$ indicates adjacency in G. Then

$$\mathrm{P}\left(\bigcap_{A \in \mathcal{A}} \bar{A}\right) > \prod_{A \in \mathcal{A}} (1 - x_A),$$

[*] Courant Institute of Mathematical Sciences, New York University, 251 Mercer St, Rm 921, New York, NY 10012 Email: pegden@math.nyu.edu. Partially supported by NSF MSPRF grant 1004696.

and so in particular, we have

$$\mathrm{P}\left(\bigcap_{A\in\mathcal{A}} \bar{A}\right) > 0. \tag{2}$$

The first breakthrough in finding an algorithmic version of the Local Lemma was made by Beck, who demonstrated his method on the classical Local Lemma application to 2-colorable hypergraphs. Beck's method was subsequently refined and given a more general framework [1,4,9,14], but required stronger bounds on the probabilities of the events than were required by the nonalgorithmic version.

In Moser and Tardos' recent breakthrough paper [10], they give an algorithmic proof of the Lovász Local Lemma in a setting which is general enough for nearly all applications of the Lemma in combinatorics, with bounds identical to those required by the nonalgorithmic version. In the framework Moser and Tardos consider, the events in $\mathcal{A}$ depend on some underlying set $\mathcal{V}$ of independent random variables, and they denote by $\mathrm{vbl}(A)$ ($A \in \mathcal{A}$) the minimal set of random variables from $\mathcal{V}$ on which each A depends; A is said to be 'violated' with respect to a particular evalation of the variables in $\mathrm{vbl}(A)$ if the event occurs for that evaluation. A Moser-Tardos dependency graph is one which implies that if events A and B are nonadjacent, then $\mathrm{vbl}(A)$ is disjoint from $\mathrm{vbl}(B)$. (Note that this notion of a dependency graph is more restrictive than the Lovász version based on probabilistic independence, as is demonstrated by an example of Kolipaka and Szegedy [8].) Moser and Tardos's theorem is then the following:

Theorem 1.2 (Moser and Tardos)**.** *Let $\mathcal{V}$ be a finite set of mutually independent variables in a probability space, and let $\mathcal{A}$ be a finite family of events determined by these variables. If there are real numbers $0 < x_A < 1$ ($A \in \mathcal{A}$) such that*

$$\mathrm{P}(A) \le x_A \prod_{B\sim A}(1 - x_B) \tag{3}$$

then there exists an assignment to the variables $\mathcal{V}$ which corresponds to no occurrence of any event from $\mathcal{A}$. Moreover, the randomized algorithm described below resamples an event A at most an expected $\frac{x_A}{1-x_A}$ times before finding the evaluation, thus the total number of resampling steps is $\sum_{A\in\mathcal{A}} \frac{x_A}{1-x_A}$ in expectation.

The Moser-Tardos algorithm consists just of beginning with a random evaluation of all the variables in $\mathcal{V}$, and then resampling $\mathrm{vbl}(A)$ for any violated events A until no violated events remain. Of course, the efficiency of the algorithm depends on the ability to resample variables efficiently and check whether individual events are violated; this is generally an easy implementation problem, however, making the analysis of the number of resampling steps the important issue.

Recently, Bissacot, Fernández, Procacci and Scoppola proved the following improvement of the Lovász Local Lemma:

Theorem 1.3 (Bissacot, et al. [3])**.** *Consider a finite family $\mathcal{A}$ of events in some probability space Ω, with some dependency graph G. If there are real numbers $0 < \mu_A < \infty$ such that*

$$P(A) \leq \frac{\mu_A}{\displaystyle\sum_{\substack{I \subset \bar{\Gamma}(A) \\ I \text{ indep.}}} \prod_{B \in I} \mu_B}, \tag{4}$$

where $\bar{\Gamma}(A) = \{A\} \cup \{B \,|\, B \sim A\}$, then $P\left(\bigcap\limits_{A \in \mathcal{A}} \bar{A} \right) > 0$.

It is not difficult to check that condition (4) is weaker than condition (1) by considering the substitution $\mu_A = \frac{x_A}{1-x_A}$. (Condition (1) would be equivalent to (4) without the condition in the sum that the sets I be independent.) In [3], they also give examples where this theorem improves some classical theorems proved with the Local Lemma. Theorem 1.3 has also since been applied to improve some theorems on graph colorings in [11].

Their proof of Theorem 1.3 is based on Shearer's characterization of labeled dependency graphs to which the conclusion of the Local Lemma applies [13] and two of those authors' recent results on the radius of convergence of logs of partition functions [7]. (The connection between the Local Lemma and the partition functions of statistical mechanics was first made by Scott and Sokal [12].)

In this short note, we prove an algorithmic analog to the result of Bissacot, et. al. That is, we will show that in the setting of Moser and Tardos's algorithmic Local Lemma, Moser and Tardos's bounds on the running time of their algorithm hold even with their condition (3) replaced with condition (4):

Theorem 1.4. *Let $\mathcal{V}$ be a finite set of mutually independent variables in a probability space, and let $\mathcal{A}$ be a finite family of events determined by these variables. If there are real numbers $0 < \mu_A < \infty$ $(A \in \mathcal{A})$ such that*

$$P(A) \leq \frac{\mu_A}{\displaystyle\sum_{\substack{I \subset \bar{\Gamma}(A) \\ I \text{ indep.}}} \prod_{B \in I} \mu_B}, \tag{5}$$

then there exists an assignment to the variables $\mathcal{V}$ which corresponds to no occurrence of any event from $\mathcal{A}$. Moreover, the Moser-Tardos algorithm resamples an event A at most an expected μ_A times before finding the evaluation, thus the total number of resampling steps is $\sum\limits_{A \in \mathcal{A}} \mu_A$ in expectation.

(The running time bound here is equivalent to the Moser-Tardos bound under the substitution $\mu_A = \frac{x_A}{1-x_A}$.)

The proof of Theorem 1.4 consists simply of re-doing one part of the proof of Moser and Tardos's theorem, taking advantage of some constraints which were not necessary for Moser and Tardos's original result.

Theorem 1.4 can be seen as doing two things: first, it extends the result of Moser and Tardos by giving a weaker condition under which the identical result holds—note that this has also been done in a more general sense by Kolipaka and Szegedy [8], who directly connect Shearer's condition with the Moser/Tardos algorithmic framework. Secondly, it gives an alternative proof of the result of Bissacot, et al. (in the slightly more restrictive algorithmic setting) which is independent of Shearer's theorem and the cluster expansion methods used in [7].

Bissacot, et al. note that their Theorem 1.3 can be extended to *Lopsided dependency graphs*, first considered by Erdős and Spencer in [6]. In their paper on their algorithmic Local Lemma, Moser and Tardos define an analog of lopsidependency in the algorithm/variable setting, and a reader familiar with Moser and Tardos's paper can easily verify that our improvement to Moser and Tardos's theorem applies to their theorem on algorithmic lopsided dependency graphs as well, as we only re-do their branching argument, which is applied to the lopsided case in the same way as in their main result.

2 Proof

The Moser-Tardos algorithm is as follows:

```
1:  procedure Moser-Tardos($\mathcal{P}$)
2:      for all $P \in \mathcal{P}$ do
3:          $v_P \leftarrow$(random evaluation of $P$)
4:      end for
5:      while $\exists A$ s.t. $A$ is violated when $P = v_P$ $(\forall P)$ do
6:          for all $P \in \mathrm{vbl}(A)$ do
7:              $v_P \leftarrow$(new random evaluation of $P$)
8:          end for
9:      end while
10: end procedure
```

(Note that when multiple events exist satisfying line 5, one of the satisfying events is chosen arbitrarily.)

Moser and Tardos' proof that this algorithm terminates in polynomial time (under condition (3)) is based on the notion of a 'witness tree'. As the algorithm runs and bad events are found and resampled, a witness tree is assigned to each step of the algorithm (where a step consists of a resampling of an event). A witness tree is a rooted tree with labels from $\mathcal{A}$. The witness tree W_t for step t of the algorithm is constructed as follows: choose as its root a vertex labeled with whatever event A_0 was resampled at step t. If the event A_1 which was resampled at step $t-1$ overlaps the label of the root, a vertex is added as a child of the root labeled with A_1. (We may have $A_1 = A_0$.) In general, for each step $i = t-1, t-2, \ldots, 1$ of the algorithm, if the event A_i which was added at step i overlaps any of the events currently labels of vertices of our partially constructed W_t, we add a vertex labeled with A_i as the child of a vertex of maximum depth whose label overlaps A_i. In the result, W_t, children always overlap their parents, and children of a common parent always get distinct labels (otherwise, whichever was added after the other would have been added as a child of the other). Any tree T with labels from $\mathcal{A}$ with these two properties is called a *proper witness tree.*

Moser and Tardos's proof of their algorithm's efficiency consists of two parts: first, they show that any proper witness tree T has probability at most

$$\prod_{v \in T} \mathrm{P}(A_v) \tag{6}$$

of occurring as a witness tree at any point in the running of the algorithm, where here A_v denotes the event labeling the vertex v.

Now, if an event A is resampled at step t, the number of occurrences of A as a label in the witness tree W_t is equal to the number of times A has been resampled on steps $1, \ldots, t$—in particular, all witness trees which will occur in a run of the algorithm will be distinct. Thus if we let $\mathcal{T}_A$ denote the set of proper witness trees with root label A, the expected value of the number N_A of resamplings of A which occur in a run of the algorithm is equal to

$$\mathrm{E}(N_A) = \sum_{T \in \mathcal{T}_A} \mathrm{P}(T \text{ occurs in the log}) \leq \sum_{T \in \mathcal{T}_A} \prod_{v \in T} \mathrm{P}(A_v). \tag{7}$$

The second part of Moser and Tardos's proof consists of bounding the sum of products in line (7). They do this by considering a random process for constructing trees: Suppose x_A ($A \in \mathcal{A}$) are real numbers between 0 and 1. Fix now any event A_0. In the first round of the process, a vertex labeled A_0 is created. In each subsequent round, for each event vertex v with label A_v created in the previous round, and for each event $A_u \in \bar{\Gamma}(A_v)$ (in the dependency graph), a vertex u with label A_u is added as a child of v with probability x_{A_u}. (All of these choices are made independently.)

Moser and Tardos prove:

Lemma 2.1 (Moser Tardos Branching Lemma)**.** *For any proper witness tree T with root labeled A_0, the probability p_T that the process above produces exactly the tree T is*

$$p_T = \frac{1 - x_{A_0}}{x_{A_0}} \prod_{v \in T} \left(x_{A_v} \prod_{B \sim A_v} (1 - x_B) \right). \tag{8}$$

□

Thus, the Lemma gives us that

$$1 \geq \sum_{T \in \mathcal{T}_A} p_T \geq \frac{1 - x_A}{x_A} \sum_{T \in \mathcal{T}_A} \prod_{v \in T} \left(x_{A_v} \prod_{B \sim A_v} (1 - x_B) \right) \tag{9}$$

Thus the bound $P(A) \leq (x_A \prod_{B \sim A} (1 - x_B))$ for all A implies, together with line (7), that

$$\mathrm{E}(N_A) \leq \frac{x_A}{1 - x_A}. \tag{10}$$

Our improvement comes just from a slightly more careful branching argument. Note that any witness tree which occurs in the log of the algorithm has the property that any children of a common vertex have labels which are nonadjacent in the dependency graph. This condition—let's call it *strongly proper*—is stronger than requiring simply that children be distinct. Thus, we can strengthen line 7, as we have the bound

$$\mathrm{E}(N_A) = \sum_{T\in\mathcal{T}_A^S} \mathrm{P}(T \text{ occurs in the log}) \le \sum_{T\in\mathcal{T}_A^S}\prod_{v\in T}\mathrm{P}(A_v). \tag{11}$$

where $\mathcal{T}_A^S \subset \mathcal{T}_A$ is the set of strongly proper witness trees.

To bound the sum in (11), we consider a modified branching process which proceeds as follows.

Given real numbers $0 < \mu_A < \infty$, we define $x_A = \frac{\mu_A}{\mu_A+1}$ (note that $0 < x_A < 1$) and fix any event A_0. In the first round of the process, a vertex labeled A_0 is created. In each subsequent round, for each event vertex v with label A_v in the previous round, we carry out a 'subprocess', where for each $A_u \in \bar{\Gamma}(v)$ (in the dependency graph), a vertex u with label A_u is added as a child of v with probability x_{A_u} (the choices are independent). At the end of the subprocess, we check if the label-set of the resulting set of children for v is an independent set in the dependency graph. If it is not, we delete the children created and restart the subprocess. Note that $x_A < 1$ (for all A) implies that the subprocess will eventually end (with probability 1) having produced an independent set.

Note that the process described above is equivalent to one in which, in each round and for each vertex v from the previous round, we create a set of children u with labels from a set chosen from all independent sets $I_v \subset \bar{\Gamma}(v)$, where the likelihood of the choice of each independent set I_v is weighted according the the product

$$w(I_v) = \left(\prod_{u\in I_v} x_{A_u}\right)\left(\prod_{u\in\bar{\Gamma}(v)\setminus I_v}(1-x_{A_u})\right).$$

Lemma 2.2 (Improved Branching Lemma). *For any strongly proper witness tree T with root labeled A_0, the probability p'_T that the modified branching process described above produces exactly the tree T is*

$$p'_T = \mu_{A_0}^{-1}\prod_{v\in T}\frac{\mu_{A_u}}{\displaystyle\sum_{\substack{I\subset\bar{\Gamma}(A_v)\\ I \text{ indep.}}}\prod_{A\in I}\mu_A}. \tag{12}$$

Proof. Letting $W_v = \bar{\Gamma}_G(A_v)\setminus \ell(\Gamma_T^+(v))$, where $\Gamma_T^+(v)$ is the set of children of v in T, and $\ell(\Gamma_T^+(v))$ is the set of their labels, we have

$$p'_T = \prod_{v\in T}\frac{\displaystyle\prod_{u\in\Gamma_T^+(v)} x_{A_u}\prod_{B\in W_v}(1-x_B)}{\displaystyle\sum_{\substack{I\subset\bar{\Gamma}(A_v)\\ I \text{ indep.}}}\prod_{A\in I}x_A\prod_{B\in\bar{\Gamma}_G(A_v)\setminus I}(1-x_B)}.$$

This can be rewritten as

$$p'_T = \prod_{v\in T} \frac{\displaystyle\prod_{u\in\Gamma_T^+(v)} \frac{x_{A_u}}{1-x_{A_u}}}{\displaystyle\sum_{\substack{I\subset\bar{\Gamma}(A_v)\\ I \text{ indep.}}} \prod_{A\in I} \frac{x_A}{1-x_A}}$$

by dividing the top and bottom by $\prod_{B\in\bar{\Gamma}_G(A_v)}(1-x_B)$. Since taking the double product $\prod_{v\in T}\prod_{u\in\Gamma_T^+(v)}$ is equivalent to taking a product $\prod_{v\in T\setminus\{v_0\}}$, where v_0 denotes the root vertex of T, this gives line (12), recalling that $x_A = \frac{\mu_A}{\mu_A+1}$ and so $\frac{x_A}{1-x_A} = \mu_A$. □

This is applied now in the same way as the branching lemma used by Moser and Tardos, but with regards to the family $\mathcal{T}_A^S$ of strongly proper witness trees rooted with A, instead of the family $\mathcal{T}_A$ of proper witness trees rooted with A. We have

$$1 \geq \sum_{T\in\mathcal{T}_A^S} p'_T \geq \mu_{A_0}^{-1} \sum_{T\in\mathcal{T}_A^S} \prod_{v\in T} \frac{\mu_{A_u}}{\displaystyle\sum_{\substack{I\subset\bar{\Gamma}(A_v)\\ I \text{ indep.}}} \prod_{A\in I} \mu_A}. \tag{13}$$

Putting this together with line (11), we see then that the condition (5) of the theorem implies that

$$\mathrm{E}(N_A) \leq \mu_A, \tag{14}$$

completing the proof the the Moser-Tardos algorithm still terminates in expected time

$$\sum_{A\in\mathcal{A}} \mu_A.$$

□

Acknowledgement

I'd like to thank Joel Spencer for some helpful discussions on this note.

References

[1] N. Alon. A parallel algorithmic version of the local lemma, *Random Structures and Algorithms* **2** (1991) 367378.

[2] Jószef Beck. An Algorithmic Approach to the Lovász Local Lemma, *Random Structures and Algorithms* **2** (1991) 343–365.

[3] R. Bissacot, R. Fernández, A. Procacci, B. Scoppola. Title: An Improvement of the Lovsz Local Lemma via Cluster Expansion. `http://arxiv.org/abs/0910.1824v2` (10 pages).

[4] A. Czumaj and C. Scheideler. Coloring non-uniform hypergraphs: a new algorithmic approach to the general Lovász local lemma, *Symposium on Discrete Algorithms* (2000) 3039.

[5] P. Erdős and L. Lovász. Problems and results on 3-chromatic hypergraphs and some related questions, in *Infinite and Finite sets* **II**, North-Holland, pp. 609627, A. Hajnal, R. Rado, and V. T. Sós (Eds.) (1975).

[6] P. Erdős, J. Spencer. Lopsided Lovász Local Lemma and Latin transversals, *Discrete Applied Math.* **30** (1991) 151–154.

[7] R. Fernandez, A. Procacci. Cluster expansion for abstract polymer models: New bounds from an old approach. *Communications in Mathematical Physics* **274** (2007) 123–140 (2007).

[8] K. Kolipaka, M. Szegedy. Moser and Tardos meet Lovász. Manuscript (2010).

[9] M. Molloy and B. Reed. Further Algorithmic Aspects of the Local Lemma, *Proceedings of the 30th Annual ACM Symposium on the Theory of Computing* (1998) 524529.

[10] R. Moser and G. Tardos. A constructive proof of the general Lovász Local Lemma, *J. ACM* **57** (2010) Article 11, 15 pages.

[11] S. Ndreca, A. Procacci, B. Scoppola. Improved bounds on coloring of graphs, `http://arxiv.org/abs/1005.1875`.

[12] A. Scott and A. Sokal. The repulsive lattice gas, the independent-set polynomial, and the Lovász local lemma, *J. Stat. Phys.* **118** (2005) 11511261.

[13] J. Shearer. On a problem of Spencer, *Combinatorica* **5** (1985), 241-245.

[14] A. Srinivasan. Improved algorithmic versions of the Lovász Local Lemma, *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms (SODA)* (2008) 611620.