The Lovász Local Lemma & its Consequences Part: 2

Usage & Applications

A presentation by -

Soham Mallick

Symposium on Combinatorics & Probability

30th April, 2022

Outline:

1. **$k-$CNF**:

   Its similarity to the hypergraph coloring problem.

   Theoretical Setup for the Algorithm

2. **The Moser-Tardos Algorithm:**

   The Algorithm

   Analysis and Proof of Correctness

   Parallelism

3. **Further Applications:**

   Packet Routing

$k-$CNF

Conjunctive Normal Form:

$$\phi = \mathbf{c_1} \wedge \mathbf{c_2} \wedge \cdots \wedge \mathbf{c_m}$$

Each $\mathbf{c_i}$ is a disjunction of **k** many boolean literals.

To determine: **If ϕ is satisfiable**

$k-$CNF

$k-$CNF

$k-$CNF

Theorem:

If each X_i occurs in at most $\frac{2^k}{ek}$ many clauses, then ϕ is satisfiable.

Theoretical Setup

Let $\mathbf{X_i} : \Omega \to \mathbb{R}, \quad i \in [n]$ independent

Let $\chi = \{X_1,\ X_2,\ \dots\ X_n\}$

Denote the bad events as $\mathbf{E_1}, \mathbf{E_2}, \dots, \mathbf{E_m}$. Each $\mathbf{E_i}$ can be completely determined by $\texttt{vbl}(\mathbf{E_i}) = \{\mathbf{X_{i_1}},\ \mathbf{X_{i_2}},\ \dots \mathbf{X_{i_j}}\}$

Dependency Graph ?

Theoretical Setup

The Moser-Tardos Algorithm

The Moser-Tardos Algorithm

Algorithm: LLL_CONSTRUCT(χ, E_i's)

for *every* $X \in \chi$ **do**
    Sample X
end for

while *there are violated events* **do**
    Pick a violated event E
    for *every* $X \in \texttt{vbl}(E)$ **do**
        Sample X
    end for
end while

return the final assignment of X_i's

Runtime of the Algorithm

$$\mathbb{E}(\#\text{ of resamplings }) \leq \sum_{i=1}^{n} \frac{x(E_i)}{1 - x(E_i)}$$

Analysis of the Algorithm

1. Log of Execution **C**
2. Witness Tree `wtree(t)`
3. Properties
4. Probability of *occurrence* of `wtree`
5. Branching Process

Witness Tree Creation

Properties of `wtree(t)`

Given log C at time t, the witness tree wtree(t) has the following properties:

(i) If $r < s, A_r$ is added to `wtree(t)` at time r and A_s added to `wtree(t)` at time s, and A_r and A_s overlap, then

$$\textbf{depth of } A_r > \textbf{ depth of } A_s$$

(ii) All the vertices at the same level in `wtree(t)` are **independent**

(iii) All trees up to time t are **different**; `wtree(1)` $\neq$ `wtree(2)` $\neq \cdots \neq$ `wtree(t)`.

Properties

Probability of occurrence of `wtree`

Probability of occurrence of `wtree`

$$\mathbb{P}(\texttt{wtree occurs in } C) = \prod_{E_i \in V(\texttt{wtree})} \mathbb{P}(E_i)$$

$$R_{E_i} = \sum_{\text{wtree } \in \mathcal{T}_{E_i}} 1_{\{\texttt{wtree occurs in } C\}}.$$

Thus,

$$\begin{aligned}
\mathbb{E}\left(R_{E_i}\right) &= \sum_{\texttt{wtree} \in \mathcal{T}_{E_i}} \mathbb{P}(\texttt{wtree occurs in } C) \\
&= \sum_{\texttt{wtree} \in \mathcal{T}_{E_i}} \prod_{E_j \in V(\texttt{wtree})} \mathbb{P}\left(E_j\right) \\
&\leqslant \sum_{\texttt{wtree} \in \mathcal{T}_{E_i}} \prod_{E_j \in V(\texttt{wtree})} x\left(E_j\right) \prod_{E_k \in N(E_j)} \left(1 - x\left(E_k\right)\right)
\end{aligned}$$

Branching Process

Branching Process

Result:

$$\mathbb{P}\left(\text{wtree}_{GW} = \text{wtree}\right) = \frac{1 - x\left(E_i\right)}{x\left(E_i\right)} \prod_{E_j \in V(\text{wtree})} x\left(E_j\right) \prod_{E_k \in N\left(E_j\right)} \left(1 - x\left(E_k\right)\right)$$

Branching Process

Parallelism

Instead of picking **one** violated event at a time, the algorithm greedily picks a **maximal independent set** of violated events in the dependency graph G, at every resampling step.

Benefit:

If the bound of LLL is weakened only a little, then

$$\mathbb{E}(\#\text{ of resamplings }) = O\left(\frac{1}{\epsilon}\log\left(\sum_{i=1}^{n}\frac{x(E_i)}{1-x(E_i)}\right)\right)$$

Packet Scheduling in Networks

$\mathbf{P_1}, \mathbf{P_2}, \ldots \mathbf{P_r}$: directed paths in a network

Design a packet scheduling policy to **minimise** time

Notations:

1. c: (Congestion) is the maximum number of paths that share a directed edge
2. d: (Dilation) is the maximum path length that occurs in the path system.

Packet Scheduling in Networks continued. . .

Observation:

$$T \geq \max\{c,\ d\}$$

Result: (Leighton, Maggs & Richa)

In the above model, there is always a packet schedule that acheives $O(c+d)$ delivery time for each packet.