Frugal Colouring of Graphs with Girth At Least Five

Ziyang Jin[1]

[1]Theory Group
Department of Computer Science
University of Toronto

Table of Contents

Frugal Colouring

A **proper vertex colouring** of a graph is an assignment of colours to each vertex in the graph such that no two adjacent vertices get the same colour.

Frugal colouring was first introduced in [Hind-Molloy-Reed '97]:

Definition (Frugal Colouring)

We say a proper vertex colouring of a graph G is β-**frugal** if, for every vertex v, no colour is assigned to more than β vertices in the neighbourhood of v.

$N(v)$: the neighbourhood of v, i.e., the set of vertices adjacent to v.

Why Frugal Colouring?

- Frugal colouring was introduced to help obtain a *total colouring* of a graph. Every graph with maximum degree Δ has a $\Delta + O(\log^8 \Delta)$ total colouring, by beginning with an $O(\log^8 \Delta)$ frugal colouring [Hind-Molloy-Reed '98].
- Frugal colouring of planar graphs is a generalization of the problem of bounding the chromatic number of the square of a planar graph [Amini-Esperet-van den Heuvel '07].
- Frugal colouring is also closely related to other types of colouring, such as linear colouring [Yuster '97].

The Main Theorem

Definition (Girth)

The **girth** of a graph is the length of a shortest cycle contained in the graph.

Here is the main theorem:

Theorem (Main Theorem)

For any graph G with girth at least five and maximum degree Δ, there exists an $O(\log^2 \Delta)$-frugal colouring using $(1 + o(1))\frac{\Delta}{\ln \Delta}$ colours.

History and the Natural Lower Bound

Chromatic Number

[Brooks' Theorem '41] Any graph G with maximum degree Δ has $\chi(G) \leq \Delta + 1$.

Frugality

[Molloy-Reed '09] $O(\log \Delta / \log \log \Delta)$-frugal.

UNIVERSITY OF TORONTO

History and the Natural Lower Bound

Chromatic Number

[Brooks' Theorem '41] Any graph G with maximum degree Δ has $\chi(G) \leq \Delta + 1$.

[Kim '95] Any graph G with maximum degree Δ and girth at least five has $\chi(G) \leq (1 + o(1))\frac{\Delta}{\ln \Delta}$.

Frugality

[Molloy-Reed '09] $O(\log \Delta / \log\log \Delta)$-frugal.

Lower bound is $\Omega(\log \Delta)$-frugal.
We prove $O(\log^2 \Delta)$-frugal.

History and the Natural Lower Bound

Chromatic Number

[Brooks' Theorem '41] Any graph G with maximum degree Δ has $\chi(G) \leq \Delta + 1$.

[Kim '95] Any graph G with maximum degree Δ and girth at least five has $\chi(G) \leq (1 + o(1))\frac{\Delta}{\ln \Delta}$.

[Molloy '19] Any triangle-free graph G with maximum degree Δ has $\chi(G) \leq (1 + o(1))\frac{\Delta}{\ln \Delta}$.

Frugality

[Molloy-Reed '09] $O(\log \Delta / \log \log \Delta)$-frugal.

Lower bound is $\Omega(\log \Delta)$-frugal.

We prove $O(\log^2 \Delta)$-frugal.

We conjecture $O(\log \Delta)$-frugal.

How to prove this?

How to prove this?

We use the probabilistic method: Design a random experiment, analyse the random experiment, and prove the wanted frugal colouring exists with positive probability.

The Naïve Colouring Procedure

Independently assign each vertex a colour:

- Each vertex v keeps track of a list of colours $L_v = \{1,\ldots,C\}$. Whenever we need to assign a colour to v, we choose a colour uniformly at random from L_v.
- If v is assigned a colour c, then we remove c from L_u for all vertices u in the neighbourhood of v.
- For two adjacent vertices u, v, if they both get assigned the same colour, we uncolour both of them.

The Naïve Colouring Procedure

Independently assign each vertex a colour:

- Each vertex v keeps track of a list of colours $L_v = \{1,...,C\}$. Whenever we need to assign a colour to v, we choose a colour uniformly at random from L_v.
- If v is assigned a colour c, then we remove c from L_u for all vertices u in the neighbourhood of v.
- For two adjacent vertices u, v, if they both get assigned the same colour, we uncolour both of them.

☹ This does not work for colouring graphs with girth at least 5 using $C = (1 + o(1))\frac{\Delta}{\ln \Delta}$ colours, because too many vertices get uncoloured.

The Naïve Colouring Procedure

Independently assign each vertex a colour:

- Each vertex v keeps track of a list of colours $L_v = \{1,...,C\}$. Whenever we need to assign a colour to v, we choose a colour uniformly at random from L_v.
- If v is assigned a colour c, then we remove c from L_u for all vertices u in the neighbourhood of v.
- For two adjacent vertices u, v, if they both get assigned the same colour, we uncolour both of them.

☹ This does not work for colouring graphs with girth at least 5 using $C = (1 + o(1))\frac{\Delta}{\ln \Delta}$ colours, because too many vertices get uncoloured.

To fix it, we colour the graph iteratively by assigning colours to a subset of vertices in each iteration.

Semi-random Method

Definition (Semi-random Method)

We construct an object X with the desired combinatorial property via a series of partial objects $X_1, X_2, ..., X_t = X$. At each step, we prove the existence of an extension of X_i to a suitable X_{i+1} by considering a random choice for that extension and applying the probabilistic method.

Semi-random method is also known as "pseudo-random method", or "Rödl Nibble" [Rödl '85].

Random Colouring Procedure (ith iteration)

1. For each vertex v, truncate L_v by removing the largest colours such that $|L_v| = L_i$ (to be defined later).
2. For each uncoloured vertex v, *activate* v with probability $\frac{K}{\ln \Delta}$, where K is a small constant.
3. For each activated vertex v, assign a random colour from L_v.
4. For each vertex v that has been assigned a colour c in the previous step, remove c from L_u for every $u \in N(v)$.
5. Simultaneously uncolour all vertices that receive the same colour as a neighbour in step 3.
6. For each vertex v and for each colour $c \in L_v$, conduct an *equalizing coin flip* (to be specified later). Remove colour c from L_v if it loses the coin flip.

The Colouring Algorithm

Algorithm 1 Random Colouring Algorithm

Require: G is a Δ-regular graph with girth five, and each vertex has no colour assigned yet

$C \leftarrow \lceil (1+\epsilon)\frac{\Delta}{\ln \Delta} \rceil$;
For each vertex v, set its colour list $L_v \leftarrow \{1, ..., C\}$;
$i \leftarrow 1$;
while Termination Condition is false **do**
    if Property P(i) is true **and** Property F($i-1$) is true **then**
        Execute Random Colouring Procedure(i);
        $i \leftarrow i + 1$;
    else
        Abort and output `fail`;
    end if
end while
Execute The Finishing Blow;

Tracking Parameters

$l_i(v)$: The size of L_v at the beginning of iteration i.
$t_i(v)$: The number of uncoloured neighbours of v at the beginning of iteration i.

We win if after some iterations, $l_i(v) \geq t_i(v) + 1$ for every vertex v.

Tracking Parameters

$l_i(v)$: The size of L_v at the beginning of iteration i.
$t_i(v)$: The number of uncoloured neighbours of v at the beginning of iteration i.

We win if after some iterations, $l_i(v) \geq t_i(v) + 1$ for every vertex v.

However, this also does not work! ☹ $t_i(v)$ drops slower than $l_i(v)$ so this never happens.

Tracking Parameters

$l_i(v)$: The size of L_v at the beginning of iteration i.
$t_i(v)$: The number of uncoloured neighbours of v at the beginning of iteration i.

We win if after some iterations, $l_i(v) \geq t_i(v) + 1$ for every vertex v.

However, this also does not work! ☹ $t_i(v)$ drops slower than $l_i(v)$ so this never happens.

☻ Find better parameters that tracks the process more carefully.

Tracking Parameters

$l_i(v)$: The size of L_v at the beginning of iteration i.
$t_i(v, c)$: At the beginning of iteration i, the number of uncoloured neighbours $u \in N(v)$ where $c \in L_u$.
$\text{Keep}_i(v, c)$: The probability that no neighbour of v is assigned colour c during iteration i.

$$\text{Keep}_i(v, c) = \left(1 - \frac{K}{\ln \Delta} \times \frac{1}{L_i}\right)^{t_i(v,c)}$$

Tracking Parameters

$l_i(v)$: The size of L_v at the beginning of iteration i.
$t_i(v,c)$: At the beginning of iteration i, the number of uncoloured neighbours $u \in N(v)$ where $c \in L_u$.
$\text{Keep}_i(v,c)$: The probability that no neighbour of v is assigned colour c during iteration i.

$$\text{Keep}_i(v,c) = \left(1 - \frac{K}{\ln \Delta} \times \frac{1}{L_i}\right)^{t_i(v,c)}$$

Good, but they are still hard to track! Every vertex v has its own $l_i(v)$, and every vertex-colour pair has its own $t_i(v,c)$.

UNIVERSITY OF TORONTO

Tracking Parameters

$l_i(v)$: The size of L_v at the beginning of iteration i.
$t_i(v, c)$: At the beginning of iteration i, the number of uncoloured neighbours $u \in N(v)$ where $c \in L_u$.
$\text{Keep}_i(v, c)$: The probability that no neighbour of v is assigned colour c during iteration i.

$$\text{Keep}_i(v, c) = \left(1 - \frac{K}{\ln \Delta} \times \frac{1}{L_i}\right)^{t_i(v,c)}$$

Good, but they are still hard to track! Every vertex v has its own $l_i(v)$, and every vertex-colour pair has its own $t_i(v, c)$.

☺ Many random variables are concentrated around their expectation. So we define deterministic values around expectations of random variables.

Expectations of l_{i+1} and t_{i+1}

Let's compute the expectations of $l_{i+1}(v)$ and $t_{i+1}(v, c)$.

$$\mathbb{E}[l_{i+1}(v)] = l_i(v) \times \text{Keep}_i,$$
$$\mathbb{E}[t_{i+1}(v, c)] \approx t_i(v, c) \left(1 - \frac{K}{\ln \Delta} \text{Keep}_i\right) \text{Keep}_i.$$

The first Keep_i inside the parenthesis is essentially $\text{Keep}_i(u, c)$ for $u \in N(v)$, i.e., the probability that an activated neighbour retains its colour. The second Keep_i outside is essentially $\text{Keep}_i(v, c)$, i.e., the probability that colour c itself is retained in L_v at the end of iteration i. We use the approximation sign as the two events are not independent, but close enough.

L_i and T_i

$l_i(v)$: The size of L_v at the beginning of iteration i.
$t_i(v, c)$: At the beginning of iteration i, the number of uncoloured neighbours $u \in N(v)$ where $c \in L_u$.

Let $L_1 = (1+\epsilon)\frac{\Delta}{\ln \Delta}$ and $T_1 = \Delta$, and recursively define

$$\text{Keep}_i = \left(1 - \frac{K}{\ln \Delta} \times \frac{1}{L_i}\right)^{T_i}, \text{ (*equalizing coin flip)}$$

$$L_{i+1} = L_i \times \text{Keep}_i - L_i^{2/3},$$

$$T_{i+1} = T_i \left(1 - \frac{K}{\ln \Delta}\text{Keep}_i\right)\text{Keep}_i + T_i^{2/3}.$$

We can focus on tracking L_i and T_i if all $l_i(v) \geq L_i$ and all $t_i(v, c) \leq T_i$.

UNIVERSITY OF TORONTO

L_i' and T_i'

We need the $-L_i^{2/3}$ and $+T_i^{2/3}$ relaxation terms to buy us larger probability in concentration bounds.

$$L_{i+1} = L_i \times \mathrm{Keep}_i - L_i^{2/3},$$
$$T_{i+1} = T_i \left(1 - \frac{K}{\ln \Delta}\mathrm{Keep}_i\right)\mathrm{Keep}_i + T_i^{2/3}.$$

L'_i and T'_i

We need the $-L_i^{2/3}$ and $+T_i^{2/3}$ relaxation terms to buy us larger probability in concentration bounds.

$$L_{i+1} = L_i \times \text{Keep}_i - L_i^{2/3},$$
$$T_{i+1} = T_i \left(1 - \frac{K}{\ln \Delta}\text{Keep}_i\right)\text{Keep}_i + T_i^{2/3}.$$

☹ Still too complicated due to the relaxation terms. Ideally,

$$L'_{i+1} = L'_i \times \text{Keep}_i,$$
$$T'_{i+1} = T'_i \left(1 - \frac{K}{\ln \Delta}\text{Keep}_i\right)\text{Keep}_i.$$

L'_i and T'_i

We need the $-L_i^{2/3}$ and $+T_i^{2/3}$ relaxation terms to buy us larger probability in concentration bounds.

$$
\begin{aligned}
L_{i+1} &= L_i \times \mathrm{Keep}_i - L_i^{2/3}, \\
T_{i+1} &= T_i \left(1 - \frac{K}{\ln \Delta}\mathrm{Keep}_i\right)\mathrm{Keep}_i + T_i^{2/3}.
\end{aligned}
$$

☹ Still too complicated due to the relaxation terms. Ideally,

$$
\begin{aligned}
L'_{i+1} &= L'_i \times \mathrm{Keep}_i, \\
T'_{i+1} &= T'_i \left(1 - \frac{K}{\ln \Delta}\mathrm{Keep}_i\right)\mathrm{Keep}_i.
\end{aligned}
$$

☺ It can be proved that $L_i = \Theta(L'_i)$ and $T_i = \Theta(T'_i)$ for the number of iterations we consider.

When do we win?

L_i and T_i seem to be good parameters to track the random colouring process. When do we stop the iterations?

When do we win?

L_i and T_i seem to be good parameters to track the random colouring process. When do we stop the iterations? When the remaining uncoloured vertices can be coloured easily.

Definition (Termination Condition)

For every uncoloured vertex v and every colour c, we have

$$\begin{aligned} (a) &\quad l_i(v) \geq L_i, \text{ and} \\ (b) &\quad t_i(v, c) \leq T_i, \text{ and} \\ (c) &\quad \frac{T_i}{L_i} \leq \frac{1}{5 \ln \Delta}. \end{aligned}$$

Definition (The Finishing Blow)

For each uncoloured vertex v, independently assign a random colour from L_v.

The Finishing Blow

Lemma (The Finishing Blow)

For any graph with maximum degree Δ, if for every vertex v and colour c, we have

(a) $|L_v| \geq L,$ *and*

(b) *there are at most T vertices $u \in N(v)$ where $c \in L_u$, and*

(c) $\dfrac{T}{L} \leq \dfrac{1}{5 \ln \Delta},$

then there exists a $(\ln \Delta)$-frugal colouring of the graph.

L and T are functions of Δ.

UNIVERSITY OF TORONTO

Proof of the Finishing Blow

Let $\beta = \ln \Delta$, the frugality parameter.

- Let $A_{e,c}$ be the event that two adjacent vertices u and v are both assigned the colour c. (Edge $e = uv$ and colour $c \in L_u \cap L_v$)
- Let $B_{S,c}$ be the event that every vertex $u \in S$ is assigned colour c in the random experiment. (S be a set of $\beta + 1$ vertices that share a common neighbour v and also for every $u \in S$, $c \in L_u$)
- Let $\mathcal{A} = \{A_{e,c}\}, \mathcal{B} = \{B_{S,c}\}$, and $\mathcal{E} = \mathcal{A} \cup \mathcal{B}$.

If none of the events in $\mathcal{E}$ occur, then we have a proper and frugal colouring.

UNIVERSITY OF TORONTO

Asymmetric Local Lemma

Theorem (Asymmetric Local Lemma)

Consider a set $\mathcal{E} = \{A_1, \ldots, A_n\}$ *of (typically bad) events such that each event* $A_i \in \mathcal{E}$ *is mutually independent of* $\mathcal{E} - (\mathcal{D}_i \cup \{A_i\})$, *for some* $\mathcal{D}_i \subseteq \mathcal{E}$. *If for each* $1 \leq i \leq n$, *we have*

$$(a)\quad \Pr(A_i) \leq \frac{1}{4}, \text{ and}$$

$$(b)\quad \sum_{A_j \in \mathcal{D}_i} \Pr(A_j) \leq \frac{1}{4},$$

then with positive probability, none of the events in $\mathcal{E}$ *occur.*

Continue Proof of the Finishing Blow

For the time sake, we will just show $A_{e,c}$. L is the size of the colour list. For any vertex, each colour appears in at most T neighbours' colour lists.

- $\Pr(A_{e,c}) = \frac{1}{L^2}$.
- $\Pr(B_{S,c}) = \frac{1}{L^{\beta+1}}$.
- $A_{e,c}$'s "dependent events" in $\mathcal{A}$ is at most $2LT$.
- $A_{e,c}$'s "dependent events" in $\mathcal{B}$ is at most $2\Delta\binom{T}{\beta}L$.

Thus, using the condition $\frac{T}{L} \leq \frac{1}{5 \ln \Delta}$, we can show

$$2LT \cdot \frac{1}{L^2} + 2\Delta\binom{T}{\beta}L \cdot \frac{1}{L^{\beta+1}} \leq \frac{1}{4}$$

for sufficiently large Δ. □

The Colouring Algorithm

Algorithm 1 Random Colouring Algorithm

Require: G is a Δ-regular graph with girth five, and each vertex has no colour assigned yet

$C \leftarrow \lceil (1+\epsilon)\frac{\Delta}{\ln \Delta} \rceil$;
For each vertex v, set its colour list $L_v \leftarrow \{1, ..., C\}$;
$i \leftarrow 1$;
while Termination Condition is false **do**
    if Property P(i) is true **and** Property F($i-1$) is true **then**
        Execute Random Colouring Procedure(i);
        $i \leftarrow i + 1$;
    else
        Abort and output `fail`;
    end if
end while
Execute The Finishing Blow;

Next Steps

Definition (Property P(i))

For each vertex v and colour c,

$$l_i(v) \geq L_i,$$
$$t_i(v, c) \leq T_i.$$

Definition (Property F(i))

For each vertex v and colour c, there are at most $O(\ln \Delta / \ln \ln \Delta)$ vertices $u \in N(v)$ receiving colour c during iteration i.

The Plan

- We introduce an iterative random colouring procedure. ✓
- We use $l_i(v)$ and $t_i(v, c)$ to track the progress of the colouring. ✓
- We show that L_i and T_i are good estimate of $l_i(v)$ and $t_i(v, c)$.
- There is positive probability to go to the next iteration.
- We need $O(\ln \Delta \ln \ln \Delta)$ iterations to reach the termination condition where $\frac{T_i}{L_i} \leq \frac{1}{5 \ln \Delta}$.
- At each iteration, each colour is assigned no more than $O(\ln \Delta / \ln \ln \Delta)$ times in any neighbourhood.
- Under the termination condition, we can complete a proper colouring on the remaining vertices that is also $O(\ln \Delta)$-frugal. ✓

UNIVERSITY OF TORONTO

Main Lemmas

Lemma (expectation of l_i and t_i)

If Property P(i) *holds, then for every vertex v and colour c,*

$$(a)\ \ \mathbb{E}(l_{i+1}(v)) = l_i(v) \times \mathrm{Keep}_i, \text{ and}$$

$$(b)\ \ \mathbb{E}(t'_{i+1}(v,c)) \leq t_i(v,c)\left(1 - \frac{K}{\ln\Delta}\mathrm{Keep}_i\right)\mathrm{Keep}_i + O\left(\frac{T_i}{L_i}\right).$$

Lemma (l_i and t_i are concentrated)

If Property P(i) *holds and $L_i, T_i \geq \ln^7 \Delta$, then for every vertex v and colour c,*

$$(a)\ \ \Pr\left(|l_{i+1}(v) - \mathbb{E}(l_{i+1}(v))| > L_i^{2/3}\right) < \Delta^{-\ln\Delta}, \text{ and}$$

$$(b)\ \ \Pr\left(|t'_{i+1}(v,c) - \mathbb{E}(t'_{i+1}(v,c))| > \frac{1}{2}T_i^{2/3}\right) < \Delta^{-\ln\Delta}.$$

The Plan

- We introduce an iterative random colouring procedure. ✓
- We use $l_i(v)$ and $t_i(v, c)$ to track the progress of the colouring. ✓
- We show that L_i and T_i are good estimate of $l_i(v)$ and $t_i(v, c)$. ✓
- There is positive probability to go to the next iteration.
- We need $O(\ln \Delta \ln \ln \Delta)$ iterations to reach the termination condition where $\frac{T_i}{L_i} \leq \frac{1}{5 \ln \Delta}$.
- At each iteration, each colour is assigned no more than $O(\ln \Delta / \ln \ln \Delta)$ times in any neighbourhood.
- Under the termination condition, we can complete a proper colouring on the remaining vertices that is also $O(\ln \Delta)$-frugal. ✓

UNIVERSITY OF TORONTO

Main Lemmas

Lemma (frugality per iteration)

For any vertex v and colour c, let random variable X_c be the number of vertices in $N(v)$ that receive colour c during iteration i. If Property P(i) *holds, then we have*

$$\Pr\left(X_c \geq \frac{24 \ln \Delta}{\ln \ln \Delta}\right) < \Delta^{-5.5}.$$

UNIVERSITY OF TORONTO

Frugality Proof

Let $X_{u,c}$ be the random variable that a single vertex $u \in N(v)$ is assigned colour c during iteration i.

$$\Pr\left(X_{u,c}\right) = \frac{K}{\ln \Delta} \times \frac{1}{L_i}.$$

Let S be the set of vertices $u \in N(v)$ with $c \in L_u$, so we have

$$X_c = \sum_{u \in S} X_{u,c}.$$

Therefore, $X_c \sim \mathrm{BIN}(T_i, \frac{K}{\ln \Delta} \times \frac{1}{L_i})$. It follows from Chernoff bound that

$$\Pr\left(X_c \geq \frac{24 \ln \Delta}{\ln \ln \Delta}\right) \leq \Delta^{-5.5}.$$

The Plan

- We introduce an iterative random colouring procedure. ✓
- We use $l_i(v)$ and $t_i(v, c)$ to track the progress of the colouring. ✓
- We show that L_i and T_i are good estimate of $l_i(v)$ and $t_i(v, c)$. ✓
- There is positive probability to go to the next iteration.
- We need $O(\ln \Delta \ln \ln \Delta)$ iterations to reach the termination condition where $\frac{T_i}{L_i} \leq \frac{1}{5 \ln \Delta}$.
- At each iteration, each colour is assigned no more than $O(\ln \Delta / \ln \ln \Delta)$ times in any neighbourhood. ✓
- Under the termination condition, we can complete a proper colouring on the remaining vertices that is also $O(\ln \Delta)$-frugal. ✓

UNIVERSITY OF TORONTO

Main Lemmas

Lemma (can go to the next iteration)

For all $i \geq 1$*, if* Property $\mathrm{P}(i)$ *holds and* $L_i, T_i \geq \ln^7 \Delta$*, then with positive probability, both* Property $\mathrm{P}(i+1)$ *and* Property $\mathrm{F}(i)$ *hold at the same time.*

Lovász Local Lemma

Theorem (Lovász Local Lemma)

Consider a set $\mathcal{E}$ of (typically bad) events such that for each $A \in \mathcal{E}$:

(a) $\Pr(A) \leq p < 1$, *and*

(b) A is mutually independent of a set of all but at most d of the other events

If $4pd \leq 1$, then with positive probability, none of the events in $\mathcal{E}$ occur.

Proof of Next Iteration

Bad events (1) $l_{i+1}(v) < L_{i+1}$, (2) $t_{i+1}(v, c) > T_{i+1}$, (3) $X_c \geq \frac{24 \ln \Delta}{\ln \ln \Delta}$, with probability

$$(1)\ \Pr\left(|l_{i+1}(v) - \mathbb{E}(l_{i+1}(v))| > L_i^{2/3}\right) < \Delta^{-\ln \Delta},$$

$$(2)\ \Pr\left(|t'_{i+1}(v, c) - \mathbb{E}(t'_{i+1}(v, c))| > \frac{1}{2} T_i^{2/3}\right) < \Delta^{-\ln \Delta},$$

$$(3)\ \Pr\left(X_c \geq \frac{24 \ln \Delta}{\ln \ln \Delta}\right) < \Delta^{-5.5}.$$

Each event is mutually independent of all but at most Δ^5 other events, so $4\Delta^{-5.5}\Delta^5 \leq 1$ and $4\Delta^{-\ln \Delta}\Delta^5 \leq 1$. It follows from L.L.L. that with positive probability, none of these bad events occur.

The Plan

- We introduce an iterative random colouring procedure. ✓
- We use $l_i(v)$ and $t_i(v,c)$ to track the progress of the colouring. ✓
- We show that L_i and T_i are good estimate of $l_i(v)$ and $t_i(v,c)$. ✓
- There is positive probability to go to the next iteration. ✓
- We need $O(\ln \Delta \ln \ln \Delta)$ iterations to reach the termination condition where $\frac{T_i}{L_i} \leq \frac{1}{5 \ln \Delta}$.
- At each iteration, each colour is assigned no more than $O(\ln \Delta / \ln \ln \Delta)$ times in any neighbourhood. ✓
- Under the termination condition, we can complete a proper colouring on the remaining vertices that is also $O(\ln \Delta)$-frugal. ✓

UNIVERSITY OF TORONTO

Can Reach Termination Condition

Lemma (Can Reach Termination Condition)

There exists an $i^* = O(\ln \Delta \ln \ln \Delta)$ *such that*

(a) *For all* $1 \leq i \leq i^*$, *we have* $\frac{T_i}{L_i} \geq \frac{1}{5 \ln \Delta}$ *and* $L_i > \Delta^{2\epsilon/3}$;

(b) $\frac{T_{i^*+1}}{L_{i^*+1}} \leq \frac{1}{5 \ln \Delta}$.

Summary

- We introduce an iterative random colouring procedure. ✓
- We use $l_i(v)$ and $t_i(v, c)$ to track the progress of the colouring. ✓
- We show that L_i and T_i are good estimate of $l_i(v)$ and $t_i(v, c)$. ✓
- There is positive probability to go to the next iteration. ✓
- We need $O(\ln \Delta \ln \ln \Delta)$ iterations to reach the termination condition where $\frac{T_i}{L_i} \leq \frac{1}{5 \ln \Delta}$. ✓
- At each iteration, each colour is assigned no more than $O(\ln \Delta / \ln \ln \Delta)$ times in any neighbourhood. ✓
- Under the termination condition, we can complete a proper colouring on the remaining vertices that is also $O(\ln \Delta)$-frugal. ✓

Total frugality: $O(\ln \Delta \ln \ln \Delta) \cdot O(\ln \Delta / \ln \ln \Delta) + O(\ln \Delta) = O(\ln^2 \Delta)$.

Q & A

Questions?

Thank you

Thank you!