Distributed coloring of graphs with an optimal number of colors

E. Bamas and L. Esperet

EPFL, Lausanne
Université Grenoble Alpes, France

STACS 2019
Berlin, March 15th, 2019

Distributed algorithms

The LOCAL model:

- Graph of size n, every vertex is given a unique ID between 1 and n
- Every vertex can send messages of unlimited size to its neighbors in the graph
- Synchronous rounds
- No failure
- Infinite local computational power

Any problem is solvable in $O(n)$ rounds !

Distributed algorithms

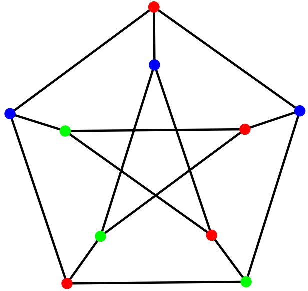
The LOCAL model:

- Graph of size n, every vertex is given a unique ID between 1 and n
- Every vertex can send messages of unlimited size to its neighbors in the graph
- Synchronous rounds
- No failure
- Infinite local computational power

Any problem is solvable in $O(n)$ rounds !

Distributed coloring, why do we care ?

- Coloring is a very powerful tool to adapt sequential algorithms to distributed model
- Nice question as many sequential coloring algorithms are challenging to adapt



Coloring graphs of maximum degree Δ

The greedy algorithm shows that every graph of maximum degree Δ can be efficiently colored with $\Delta + 1$ colors.

Coloring graphs of maximum degree Δ

The greedy algorithm shows that every graph of maximum degree Δ can be efficiently colored with $\Delta + 1$ colors.

Theorem (Brooks)

Every connected graph of maximum degree Δ, other than cliques and odd cycles, can be Δ-colored efficiently.

Coloring graphs of maximum degree Δ

The greedy algorithm shows that every graph of maximum degree Δ can be efficiently colored with $\Delta + 1$ colors.

Theorem (Brooks)

Every connected graph of maximum degree Δ, other than cliques and odd cycles, can be Δ-colored efficiently.

(it follows that deciding whether $\chi(G) \leq \Delta$ is in P, and if the answer is yes, obtaining the coloring is also in P).

Coloring graphs of maximum degree Δ

The greedy algorithm shows that every graph of maximum degree Δ can be efficiently colored with $\Delta + 1$ colors.

Theorem (Brooks)

Every connected graph of maximum degree Δ, other than cliques and odd cycles, can be Δ-colored efficiently.

(it follows that deciding whether $\chi(G) \leq \Delta$ is in P, and if the answer is yes, obtaining the coloring is also in P).

Question: can we go further?

Coloring graphs of maximum degree Δ

The greedy algorithm shows that every graph of maximum degree Δ can be efficiently colored with $\Delta + 1$ colors.

Theorem (Brooks)

Every connected graph of maximum degree Δ, other than cliques and odd cycles, can be Δ-colored efficiently.

(it follows that deciding whether $\chi(G) \leq \Delta$ is in P, and if the answer is yes, obtaining the coloring is also in P).

Question: can we go further?

No for $\Delta = 4$

Coloring graphs of maximum degree Δ

For any Δ, let k_Δ be the maximum integer k such that $(k+1)(k+2) \leq \Delta$. It can be checked that $\sqrt{\Delta} - 3 < k_\Delta < \sqrt{\Delta} - 1$.

Coloring graphs of maximum degree Δ

For any Δ, let k_Δ be the maximum integer k such that $(k+1)(k+2) \leq \Delta$. It can be checked that $\sqrt{\Delta} - 3 < k_\Delta < \sqrt{\Delta} - 1$.

Theorem (Embden-Weinert, Hougardy, Kreuter 1998)

For $3 \leq c \leq \Delta - k_\Delta - 1$, we cannot test for c-colorability of graphs with maximum degree Δ in polynomial time unless P = NP.

Coloring graphs of maximum degree Δ

For any Δ, let k_Δ be the maximum integer k such that $(k+1)(k+2) \le \Delta$. It can be checked that $\sqrt{\Delta} - 3 < k_\Delta < \sqrt{\Delta} - 1$.

Theorem (Embden-Weinert, Hougardy, Kreuter 1998)

For $3 \le c \le \Delta - k_\Delta - 1$, we cannot test for c-colorability of graphs with maximum degree Δ in polynomial time unless $\mathrm{P} = \mathrm{NP}$.

Theorem (Molloy Reed 2001–2014)

For sufficiently large (but constant) Δ, and every $c \ge \Delta - k_\Delta$, there is a linear time deterministic algorithm to test whether graphs of maximum degree Δ are c-colorable. Furthermore, there is a polynomial time deterministic algorithm that will produce a c-coloring whenever one exists.

Distributed coloring

When $\Delta = O(1)$, graphs of maximum degree Δ can be colored with $\Delta + 1$ colors in $O(\log^* n)$ rounds (SIAM J. Discrete Math. 1988, SIAM Journal on Computing 1992), and the round complexity is best possible already for paths.

Distributed coloring

When $\Delta = O(1)$, graphs of maximum degree Δ can be colored with $\Delta + 1$ colors in $O(\log^* n)$ rounds (SIAM J. Discrete Math. 1988, SIAM Journal on Computing 1992), and the round complexity is best possible already for paths.

In general, $(\Delta + 1)$-coloring is a very active field of research with the current state-of-the art running times: $O(\sqrt{\Delta \log \Delta} \log^* \Delta + \log^* n)$ deterministic complexity (FOCS 2016 + PODC 2018) and $O(\sqrt{\log \Delta}) + 2^{O(\sqrt{\log \log n})}$ randomized complexity (STOC 2016 + STOC 2018).

Distributed coloring

When $\Delta = O(1)$, graphs of maximum degree Δ can be colored with $\Delta + 1$ colors in $O(\log^* n)$ rounds (SIAM J. Discrete Math. 1988, SIAM Journal on Computing 1992), and the round complexity is best possible already for paths.

In general, $(\Delta + 1)$-coloring is a very active field of research with the current state-of-the art running times: $O(\sqrt{\Delta \log \Delta} \log^* \Delta + \log^* n)$ deterministic complexity (FOCS 2016 + PODC 2018) and $O(\sqrt{\log \Delta}) + 2^{O(\sqrt{\log \log n})}$ randomized complexity (STOC 2016 + STOC 2018).

The Δ-coloring problem (Brooks Theorem) can be solved in $O(\log \Delta) + 2^{O(\sqrt{\log \log n})}$ rounds w.h.p. when $\Delta \geq 4$, or $O((\log \log n)^2)$ rounds w.h.p. when $\Delta \geq 3$ is a constant (PODC 2018).

Our results

Recall that $k_\Delta \approx \sqrt{\Delta} - 2$.

Theorem (B. and Esperet 2018)

When $c \leq \Delta - k_\Delta$, there exist arbitrarily large graphs G of maximum degree Δ for which $\chi(G) = c$, and such that any distributed algorithm coloring G with c colors takes $\Omega(n/\Delta)$ rounds.

Our results

Recall that $k_\Delta \approx \sqrt{\Delta} - 2$.

Theorem (B. and Esperet 2018)

When $c \leq \Delta - k_\Delta$, there exist arbitrarily large graphs G of maximum degree Δ for which $\chi(G) = c$, and such that any distributed algorithm coloring G with c colors takes $\Omega(n/\Delta)$ rounds.

Theorem (B. and Esperet 2018)

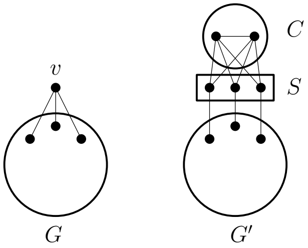
For sufficiently large Δ, there is a distributed randomized algorithm running w.h.p. in $\min\{O(\log^{1/12}(\Delta)\log n), 2^{O(\log\Delta + \sqrt{\log\log n})}\}$ rounds, that takes a graph G with maximum degree Δ in input, and outputs, for any $c \geq \Delta - k_\Delta + 1$, either a certificate that G is not c-colorable, or a c-coloring of G.

Graphs that are hard to color optimally

Recall that k_Δ is the greatest integer such that $(k_\Delta + 1)(k_\Delta + 2) \leq \Delta$ (hence $k_\Delta \approx \sqrt{\Delta} - 2$) and $c \leq \Delta - k_\Delta$

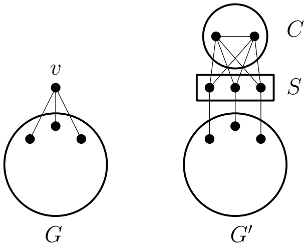
Graphs that are hard to color optimally

Recall that k_Δ is the greatest integer such that $(k_\Delta + 1)(k_\Delta + 2) \leq \Delta$ (hence $k_\Delta \approx \sqrt{\Delta} - 2$) and $c \leq \Delta - k_\Delta$



Graphs that are hard to color optimally

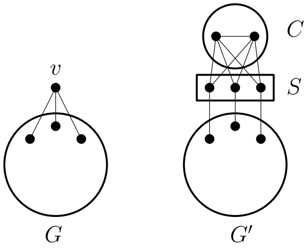
Recall that k_Δ is the greatest integer such that $(k_\Delta + 1)(k_\Delta + 2) \leq \Delta$ (hence $k_\Delta \approx \sqrt{\Delta} - 2$) and $c \leq \Delta - k_\Delta$



Set G to be a clique of size $c + 1$ and take out a vertex v.

Graphs that are hard to color optimally

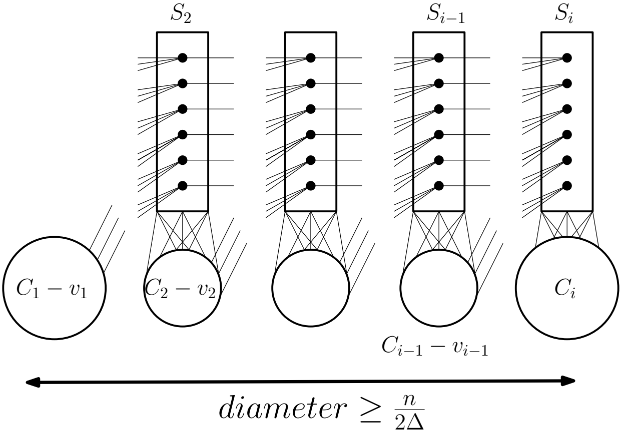
Recall that k_Δ is the greatest integer such that $(k_\Delta + 1)(k_\Delta + 2) \leq \Delta$ (hence $k_\Delta \approx \sqrt{\Delta} - 2$) and $c \leq \Delta - k_\Delta$



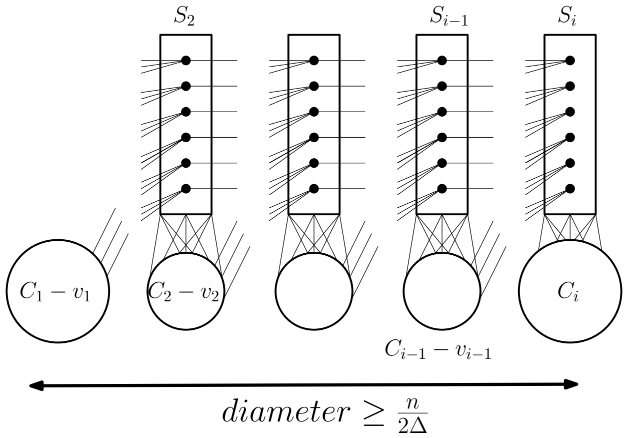
Set G to be a clique of size $c + 1$ and take out a vertex v.

Set C to be a clique of size $c - 1$ and S a stable set of size $\Delta - c + 2$. We can ensure the maximum degree is still Δ because $(\Delta - c + 1)(\Delta - c + 2) \geq \Delta$.

Graphs that are hard to color optimally



Graphs that are hard to color optimally



Theorem (B. and Esperet 2018)

When $c \leq \Delta - k_\Delta$, there exist arbitrarily large graphs G of maximum degree Δ for which $\chi(G) = c$, and such that any distributed algorithm coloring G with c colors takes $\Omega(n/\Delta)$ rounds.

Overview of the proof

We call a vertex dense if its neighborhood has more than $\binom{\Delta}{2} - \Delta^{3/2}$ edges. A vertex v that is not dense is said to be sparse.

Overview of the proof

We call a vertex dense if its neighborhood has more than $\binom{\Delta}{2} - \Delta^{3/2}$ edges. A vertex v that is not dense is said to be sparse.

We say that $S, X_1, X_2, \ldots, X_t$ is a dense decomposition of G if:

1. $S, X_1, X_2, \ldots, X_t$ partition V.
2. every X_i has between $\Delta - 8\Delta^{1/2}$ and $\Delta + 4\Delta^{1/2}$ vertices.
3. There are at most $8\Delta^{3/2}$ edges between X_i and $V - X_i$.
4. a vertex is adjacent to at least $\frac{3\Delta}{4}$ vertices of X_i if and only if it is in X_i.
5. Every vertex in S is sparse.

Overview of the proof

We call a vertex dense if its neighborhood has more than $\binom{\Delta}{2} - \Delta^{3/2}$ edges. A vertex v that is not dense is said to be sparse.

We say that $S, X_1, X_2, \ldots, X_t$ is a dense decomposition of G if:

1. $S, X_1, X_2, \ldots, X_t$ partition V.
2. every X_i has between $\Delta - 8\Delta^{1/2}$ and $\Delta + 4\Delta^{1/2}$ vertices.
3. There are at most $8\Delta^{3/2}$ edges between X_i and $V - X_i$.
4. a vertex is adjacent to at least $\frac{3\Delta}{4}$ vertices of X_i if and only if it is in X_i.
5. Every vertex in S is sparse.

Such a decomposition can be computed in a constant number of rounds.

The Distributed Lovász Local Lemma

Consider a set X of independent random variables, and a set $B = B_1, \ldots, B_n$ of (typically bad) events, each depending on a subset of the variables from X.

The Distributed Lovász Local Lemma

Consider a set X of independent random variables, and a set $B = B_1, \ldots, B_n$ of (typically bad) events, each depending on a subset of the variables from X. Consider the graph H with vertex-set B, with an edge between two events if the set of variables they depend on intersect. The graph H is called the event dependency graph.

The Distributed Lovász Local Lemma

Consider a set X of independent random variables, and a set $B = B_1, \ldots, B_n$ of (typically bad) events, each depending on a subset of the variables from X.

Consider the graph H with vertex-set B, with an edge between two events if the set of variables they depend on intersect. The graph H is called the event dependency graph.

Let $d \geq 2$ be the maximum degree of H, and let p be the maximum probability of an event from B.

The Distributed Lovász Local Lemma

Consider a set X of independent random variables, and a set $B = B_1, \ldots, B_n$ of (typically bad) events, each depending on a subset of the variables from X.
Consider the graph H with vertex-set B, with an edge between two events if the set of variables they depend on intersect. The graph H is called the event dependency graph.
Let $d \geq 2$ be the maximum degree of H, and let p be the maximum probability of an event from B.

Theorem (Chung, Pettie, Su 2014)

If $epd^2 < 1$, then there is a distributed randomized algorithm, running in H w.h.p. in $O(\log n)$ rounds, that finds a value assignment to the variables of X such that no event from B holds.

The Distributed Lovász Local Lemma

Consider a set X of independent random variables, and a set $B = B_1, \ldots, B_n$ of (typically bad) events, each depending on a subset of the variables from X. Consider the graph H with vertex-set B, with an edge between two events if the set of variables they depend on intersect. The graph H is called the event dependency graph.

Let $d \geq 2$ be the maximum degree of H, and let p be the maximum probability of an event from B.

Theorem (Chung, Pettie, Su 2014)

If $epd^2 < 1$, then there is a distributed randomized algorithm, running in H w.h.p. in $O(\log n)$ rounds, that finds a value assignment to the variables of X such that no event from B holds.

Theorem (Ghaffari, Harris, Kuhn 2017)

If $2^{15}pd^8 < 1$, then there is a distributed randomized algorithm, running in H w.h.p. in $2^{O(\log d + \sqrt{\log \log n})}$ rounds, that finds a value assignment to the variables of X such that no event from B holds.

A certificate of non-colorability

Theorem (B. and Esperet 2018)

For sufficiently large Δ, there is a distributed randomized algorithm running w.h.p. in $\min\{O(\log^{1/12}(\Delta)\log n), 2^{O(\log\Delta+\sqrt{\log\log n})}\}$ rounds, that takes a graph G with maximum degree Δ in input, and outputs, for any $c \geq \Delta - k_\Delta + 1$, either a certificate that G is not c-colorable, or a c-coloring of G.

A certificate of non-colorability

Theorem (B. and Esperet 2018)

For sufficiently large Δ, there is a distributed randomized algorithm running w.h.p. in $\min\{O(\log^{1/12}(\Delta)\log n), 2^{O(\log\Delta+\sqrt{\log\log n})}\}$ rounds, that takes a graph G with maximum degree Δ in input, and outputs, for any $c \geq \Delta - k_\Delta + 1$, either a certificate that G is not c-colorable, or a c-coloring of G.

Theorem (Molloy and Reed 2001–2014)

For sufficiently large Δ, and for $c \geq \Delta - k_\Delta + 1$, if G has maximum degree at most Δ, and $\chi(G) > c$, then there is some vertex v in G such that the subgraph induced by $\{v\} \cup N(v)$ is not c-colorable.

Overview of the algorithm

Theorem (B. and Esperet 2018)

For sufficiently large Δ, there is a distributed randomized algorithm running w.h.p. in $\min\{O(\log^{1/12}(\Delta)\log n), 2^{O(\log\Delta+\sqrt{\log\log n})}\}$ rounds, that takes a graph G with maximum degree Δ in input, and outputs, for any $c \geq \Delta - k_\Delta + 1$, either a certificate that G is not c-colorable, or a c-coloring of G.

Overview of the algorithm

Theorem (B. and Esperet 2018)

For sufficiently large Δ, there is a distributed randomized algorithm running w.h.p. in $\min\{O(\log^{1/12}(\Delta)\log n), 2^{O(\log\Delta+\sqrt{\log\log n})}\}$ rounds, that takes a graph G with maximum degree Δ in input, and outputs, for any $c \geq \Delta - k_\Delta + 1$, either a certificate that G is not c-colorable, or a c-coloring of G.

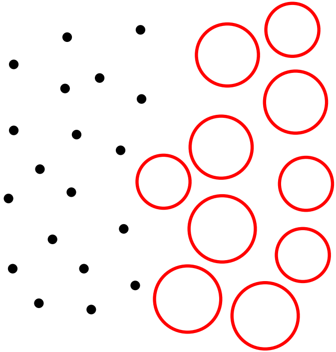
First compute a dense decomposition.

Overview of the algorithm

Theorem (B. and Esperet 2018)

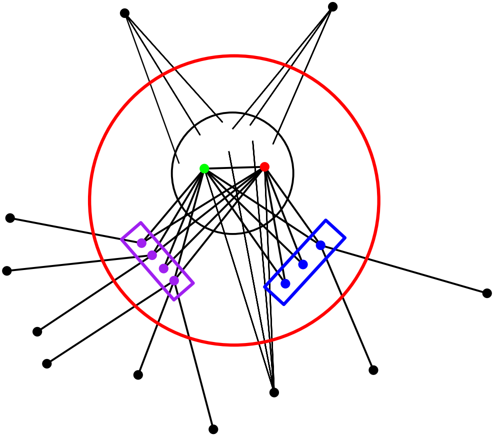
For sufficiently large Δ, there is a distributed randomized algorithm running w.h.p. in $\min\{O(\log^{1/12}(\Delta)\log n), 2^{O(\log\Delta+\sqrt{\log\log n})}\}$ rounds, that takes a graph G with maximum degree Δ in input, and outputs, for any $c \geq \Delta - k_\Delta + 1$, either a certificate that G is not c-colorable, or a c-coloring of G.

First compute a dense decomposition.



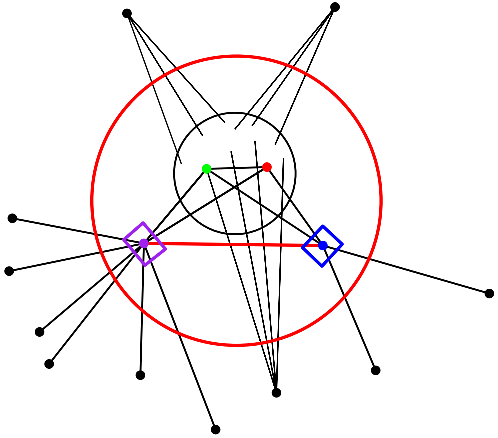
Overview of the algorithm

Build a nice c-coloring of each dense part.



Overview of the algorithm

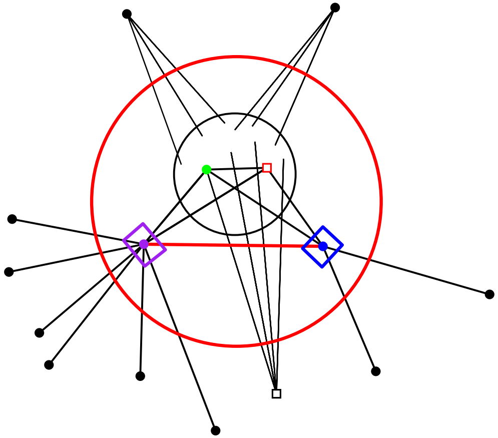
Contract each color class of size greater than 1 and add edges to make it a clique.



Overview of the algorithm

Identify external vertices that have a lot of neighbors in the dense part.

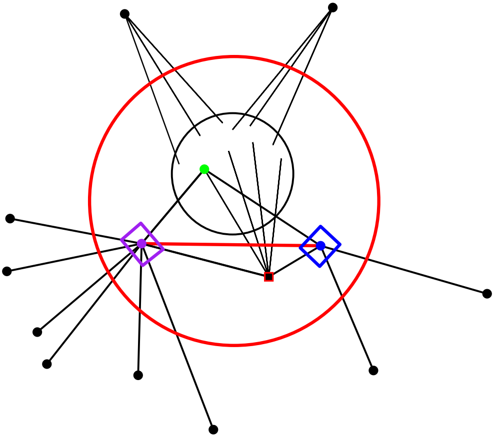
Contract these independent vertices together with an independent internal vertex.



Overview of the algorithm

Identify external vertices that have a lot of neighbors in the dense part.

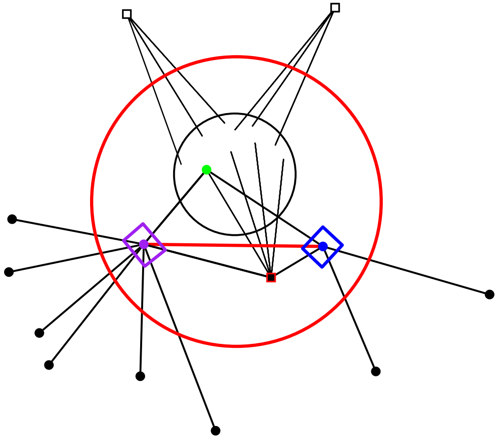
Contract these independent vertices together with an independent internal vertex.



Overview of the algorithm

Identify external vertices whose neighborhoods union cover a lot of internal vertices.

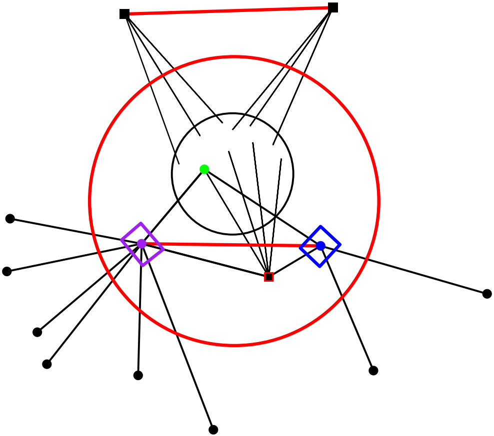
Add edges between them.



Overview of the algorithm

Identify external vertices whose neighborhoods union cover a lot of internal vertices.

Add edges between them.



Overview of the algorithm

Color the sparse part of this modified graph.

Overview of the algorithm

Color the sparse part of this modified graph.

Use repeated applications of LLL to color the dense parts. Make sure at each step that for every color, not many internal vertices have an external neighbor of this color.

Overview of the algorithm

Color the sparse part of this modified graph.

Use repeated applications of LLL to color the dense parts. Make sure at each step that for every color, not many internal vertices have an external neighbor of this color.

Deduce a valid coloring of the graph from the one on the modified graph.

Overview of the algorithm

Color the sparse part of this modified graph.

Use repeated applications of LLL to color the dense parts. Make sure at each step that for every color, not many internal vertices have an external neighbor of this color.

Deduce a valid coloring of the graph from the one on the modified graph.

Theorem (B. and Esperet 2018)

For sufficiently large Δ, there is a distributed randomized algorithm running w.h.p. in $\min\{O(\log^{1/12}(\Delta)\log n), 2^{O(\log\Delta+\sqrt{\log\log n})}\}$ rounds, that takes a graph G with maximum degree Δ in input, and outputs, for any $c \geq \Delta - k_\Delta + 1$, either a certificate that G is not c-colorable, or a c-coloring of G.

Questions

Thank you for your attention.