1 Lectures 2-3: Graph Containers

1.1 A graph container theorem

Theorem 1.1. *For every $c \geqslant 1$ and $1 > \epsilon > 0$, there exists $\delta < \frac{(1-\epsilon)\epsilon}{2c(c+\epsilon)}$ such that the following holds. Let G be a graph on n vertices, with average degree d and maximum degree at most $c \cdot d$. Then there exists a collection $\mathcal{C}$ of subsets (containers) of $V(G)$ such that*

(i) *For each independent set I, there exists a container $C \in \mathcal{C}$ such that $I \subseteq C$;*

(ii) *$|C| \leqslant (1-\delta)n$ for each $C \in \mathcal{C}$;*

(iii) *$|\mathcal{C}| \leqslant \binom{n}{\delta n/\epsilon d}$.*

Proof. The theorem will be proved by an algorithm with **input** G and I, and **output** S and $C = C(S)$, where S is the certificate of I and C depends only on S.

Step 0: We first fix an arbitrary ordering, $(v'_1, \dots, v'_n)$, of the vertex set $V(G)$. It will be used to break ties when we choose vertices in the algorithm based on other conditions. For example, when we choose max degree vertex from the graph we may have a tie, then we choose the vertex with max degree and show up first in the previously fixed ordering.

Step 1: We are given G and I. Initially set $S = \emptyset$ and $A = V(G)$, where A stands for active vertices. Let $i = 1$.

Step 2: Let v_i be the vertex of max degree in $G[A]$ chosen according to the previously fixed ordering. We check if it is true that $v_i \in I$. If **YES** then go to **Step 2.1**. If **NO** then go to **Step 2.2**.

Step 2.1: Let

$$A = A - v_i - N_{G[A]}(v_i) \quad \text{and} \quad S = S \cup \{v_i\}.$$

If $|S \cup A| > (1-\delta)n$, then go to **Step 2**. Otherwise stop and go to **Step 3**.

Step 2.2: Let

$$A = A - v_i \quad \text{and} \quad S = S.$$

Then go to **Step 2**.

Step 3: Output the final S and A. Define $S(I) = S$ and $A(I) := A$.

Note that $I \subseteq A(I) \cup S(I) =: C(S(I))$ and next we show why S is called the certificate of I and C only depends on S, not on $I - S$.

Claim 1.2. *If I_1, I_2 are two independent sets of $V(G)$ such that the outputs $S(I_1)$ and $S(I_2)$ of the algorithm are the same, then the final active sets $A(I_1)$ and $A(I_2)$ of the algorithm are also the same.*

Proof. We compare those two executions of the algorithm for I_1 and I_2. If there is a vertex $v_i \in I_1 - I_2$ (or $I_2 - I_1$ and we only show the argument for $I_1 - I_2$) that is checked before the stop of the algorithm, then $v_i \in S(I_1)$ and $v_i \notin S(I_2)$ thus $S(I_1) \neq S(I_2)$. Assume now $S(I_1) = S(I_2)$, then no such vertex (type of v_i) is checked before the end of the algorithm. Therefore, the sequence of checked vertices from the beginning to the end of the algorithm for I_1 and I_2 is the same and thus the algorithm runs exactly the same way and the outputs $A(I_1)$ and $A(I_2)$ are the same. □

Step 4: We define the collection of containers

$$\mathcal{C} := \{C(S(I)) = A(I) \cup S(I) : I \text{ independent set in } V(G)\}.$$

Claim 1.3. *For each independent set I of G, $I \subseteq C(S(I))$. Thus, condition **(i)** holds.*

Proof. In each iteration of the algorithm, we either delete a vertex that is not in I or delete a vertex v in I with its neighbours from A but added v to S. Thus, we always keep the property $I \subseteq S \cup A$. □

Claim 1.4. *In the algorithm before $|S|$ exceeds $\frac{\delta}{\epsilon d}n$, we deleted at least ϵd vertices each time we add a vertex to S.*

Proof. We assume $|S| \leqslant \frac{\delta}{\epsilon d}n$. Since the algorithm is running, we know at most δn vertices are deleted from A. Thus, for $\delta < \frac{(1-\epsilon)\epsilon}{2c(c+\epsilon)}$ and assuming $c \cdot d \geqslant 1$,

$$e(G[A]) \geqslant e(G) - (\frac{\delta}{\epsilon d}n + \delta n)\Delta(G) \geqslant e(G) - (\frac{c}{\epsilon} + 1)\delta n \cdot cd \geqslant e(G) \cdot (1 - (\frac{c}{\epsilon} + 1)2c\delta) \geqslant \epsilon e(G).$$

Therefore, $G[A]$ has average degree at least ϵd and we delete at least ϵd vertices since v_i is a max degree vertex in $G[A]$. □

Claim 1.5. *Conditions **(ii)** and **(iii)** also hold.*

Proof. Since we stop when $|S \cup A| \leqslant (1 - \delta)n$,

$$|C| = |S(I) \cup A(I)| \leqslant (1 - \delta)n.$$

We also know $|S(I)| \leqslant \frac{\delta}{\epsilon d}n$ since by Claim 1.4 deleted at least $\epsilon d \cdot \frac{\delta}{\epsilon d}n = \delta n$ vertices from A after adding vertices $\frac{\delta}{\epsilon d}n$ times and thus the algorithm stops before or at the time $|S|$ reaches $\frac{\delta}{\epsilon d}n$.

By Claim 1.2, we know

$$|\mathcal{C}| \leqslant \binom{n}{\lceil \frac{\delta}{\epsilon d}n \rceil}.$$

□

□

1.2 Other variant of Theorem 1.1

Theorem 1.6. *Let G be a graph on n vertices and $t \in R$. Then there is a collection $\mathcal{C}$ of containers satisfy that: for every independent set $I \subseteq V(G)$ there exists a pair $(S_i, C_i(S_i))$ such that $C_i(S_i) \in \mathcal{C}$ and*

1) $|S_i| \leqslant \frac{n}{t}$;

2) $\Delta(G[C_i]) < t - 1$;

3) $I \subseteq C_i$;

4) $|\mathcal{C}| \leqslant \binom{n}{\leqslant n/t}$.

An immediate corollary of Theorem 1.6 is the following statement.

Corollary 1.7. *The number of independent sets in G is upper bounded by*

$$\left(\sum_{i \leqslant \frac{n}{t}} \binom{n}{i}\right) \cdot 2^{\max_{C \in \mathcal{C}}\{|C|\}}.$$

Proof of Theorem 1.6: The theorem can be proved by an algorithm with **input** G and I, and **output** S and $C = C(S)$, where S is the certificate of I and C only depends on S.

Step 0 and **Step 1** are the same as in the proof of Theorem 1.1.

Step 2: Let v_i be the max degree vertex in $G[A]$ chosen according to the previously fixed ordering. We check if it is true that $v_i \in I$. If **YES** then go to **Step 2.1**. If **NO** then go to **Step 2.2**.

Step 2.1: Let

$$A = A - v_i - N_{G[A]}(v_i) \quad \text{and} \quad S = S \cup \{v_i\}.$$

Check if $\Delta(G[A]) \geqslant t - 1$. If **YES** then go to **Step 2**. If **NO** then go to **Step 3**.

Step 2.2: Let

$$A = A - v_i \quad \text{and} \quad S = S.$$

Then go to **Step 2**.

Step 3: Output the final S and A. Define $S(I) = S$ and $A(I) := A$.

Note that $I \subseteq A(I) \cup S(I) =: C$ and next we show why S is called the certificate of I and C only depends on S.

Claim 1.8. *If I_1, I_2 are two independent sets of $V(G)$ such that the outputs $S(I_1)$ and $S(I_2)$ of the algorithm are the same, then the final active sets $A(I_1)$ and $A(I_2)$ of the algorithm are also the same.*

Proof. Same as the proof in Theorem 1.1. □

Step 4: We define the collection of containers

$$\mathcal{C} := \{C(S(I)) : I \text{ independent set in } V(G)\}.$$

Claim 1.9. *For each independent set I of G, $I \subseteq C(S(I))$. Thus, condition **3)** holds.*

Proof. Since we only deleted vertices that are not in I and kept the property $I \subseteq S \cup A$ in each iteration of the algorithm. □

Claim 1.10. *Conditions **1)**, **2)** and **4)** also hold.*

Proof. Since we stop the algorithm when $\Delta(G[A]) < t-1$ and S is an independent set which has no neighbour in A, $\Delta(G[C]) = \Delta(G[S \cup A]) < t - 1$ and condition **2)** holds.

Since we deleted at least $1 + t - 1 = t$ vertices from A each time when we add a vertex into S, $|S| \cdot t \leqslant n$ and condition **1)** holds.

By Claim 1.8, condition **4)** holds. □

□

There are other algorithms which can prove Theorem 1.6. We show one of them. Since this variant algorithm is similar to the previous algorithm, we only show the rough idea and their difference.

Step 1: Fix an ordering $(v_1, \dots, v_n)$ of $V(G)$.

Step 2: We check each vertex u according to the ordering one by one that if $d_G(u) \geqslant r$. If **NO** then we do nothing. If **YES** then we further check if $u \in I$. If **NO** then

$$A = A - u \quad \text{and} \quad S = S.$$

If **YES** then

$$S = S \cup \{u\} \quad \text{and} \quad A = A - u - N_{G[A]}(u).$$

• Since we put vertex u in S and delete it and its neighborhood only when $d_G(u) \geqslant r$, we know

$$|S| \leqslant \frac{n}{r+1}.$$

• Since if $d_{G[A]}(u) \geqslant r$ then we would have deleted u from A in the algorithm, which is a contradiction. Thus, we conclude that

$$\Delta(G[A]) < r.$$

• We also have $I \subseteq S \cup A$, since we start with $A = V(G)$ and whenever we delete a vertex $u \in I$ from A we add it back to S.

1.3 Applications of Theorem 1.6

Example 1: Let G_n be a d-regular graph on n vertices, where $\log n \ll d \ll \frac{n}{2}$. Then

$$2^{\frac{n}{2}+\frac{n}{2d}} \leqslant \text{Number of independent sets} \leqslant 2^{\frac{n}{2}+o(n)}.$$

Proof. **Lower bound:** This is by the construction of $\frac{n}{2d}$ disjoint union of $K_{d,d}$s, assuming $2d|n$. For each copy of $K_{d,d}$, we have the freedom to pick an independent set from one of the two parts. It gives us the following lower bound:

$$(2 \cdot 2^d)^{\frac{n}{2d}} = 2^{\frac{n}{2}+\frac{n}{2d}}.$$

• Note that Jeff Kahn [3] showed the lower bound is true for bipartite graphs by the entropy method and later Yufei Zhao [4] showed that it meets the lower bound for general graphs.

Upper bound: We apply Theorem 1.6 to G_n to obtain S and C satisfying the conditions in the theorem with $|C| = \frac{n}{2} + \beta n$.

Proposition 1.11. *$\beta \to 0$ as $n \to \infty$.*

Proof. We know

$$d \cdot |C| = \sum_{v \in C} d(v) = 2e(G[C]) + e(C, \bar{C}) \leqslant 2e(G[C]) + |\bar{C}| \cdot d.$$

Thus,

$$2e(G[C]) \geqslant d(|C| - |\bar{C}|),$$

$$t \geqslant \Delta(G[C]) \geqslant \frac{d(|C| - |\bar{C}|)}{n} = 2\beta d$$

and

$$\beta \leqslant \frac{t}{2d}.$$

By the graph container theorem (Theorem 1.6), we know the number of independent sets is bounded by

$$\binom{n}{\leqslant |S|} \cdot 2^{|C|} \leqslant \binom{n}{\leqslant \frac{n}{t}} \cdot 2^{|C|} \leqslant (et)^{\frac{n}{t}} \cdot 2^{\frac{n}{2} + \frac{tn}{2d}} = 2^{\frac{n}{t}(\log t + \log e) + \frac{tn}{2d} + \frac{n}{2}}.$$

Letting $t = \sqrt{2d \log n}$, we obtain

$$2^{\frac{n}{2\sqrt{2d\log n}} \cdot (\log 2d + \log\log n) + \frac{n}{\sqrt{2d \log n}} \log e + \frac{n\sqrt{2d\log n}}{2d} + \frac{n}{2}} = 2^{\frac{n}{2} + o(n)}.$$

□

□

Example 2: Number of q-vertex-colorings[1] of regular graphs.

We begin with a warm-up example. We count the number of q-colorings of $K_{d,d}$. Since the same color cannot appear on both sides of the partition, we count the number of colorings by fixing the number of colors i appearing in one fixed side. Thus, the number of q-colorings is about

$$\sum_{i=1}^{q-1} \binom{q}{i} \cdot i^d \cdot (q-i)^d.$$

Moreover, the number of q-colorings of the vertex disjoint union of $\frac{n}{2d}$ copies of $K_{d,d}$ is

$$\left(\sum_{i=1}^{q-1} \binom{q}{i} \cdot i^d \cdot (q-i)^d \right)^{n/2d} \sim \left((q/2)^{2d} \right)^{n/2d} = (q/2)^n.$$

Conjecture 1.12 (Galvin and Tetali [2]). *Among all n-vertex, d-regular graphs with $2d|n$, none of them admits more q-colorings than the disjoint union of $n/2d$ copies of the complete bipartite graph $K_{d,d}$.*

David Galvin showed that Conjecture 1.12 is asymptotically true.

[1]Here, we focus on proper colorings and we omit the word ”proper”.

Theorem 1.13 (Galvin [1])**.** *The number of proper q-colorings admitted by an n-vertex, d-regular graph is*

$$\leqslant \begin{cases} (q^2/4)^{n/2} \cdot \binom{q}{q/2}^{\frac{n(1+o(1))}{2d}}, & \text{if } q \text{ is even} \\ ((q^2-1)/4)^{n/2} \cdot \binom{q+1}{(q+1)/2}^{\frac{n(1+o(1))}{2d}}, & \text{if } q \text{ is odd} \end{cases}$$

where $o(1) \to 0$ *as* $d \to \infty$*.*

We use the Graph Container Theorem (Theorem 1.6) to give an upper bound.

Theorem 1.14. *Let G be a d-regular n-vertex graph with $\log n \ll d \ll \frac{n}{2}$. Then the number of q-colorings is bounded by*

$$q^n \left(\frac{1}{2} + o(1)\right)^n.$$

Observation: Each color class is an independent set.

Proof. Since a q-coloring of G is equivalent to a partition of $V(G)$ into q independent sets $\{I_1, \ldots, I_q\}$, where I_i corresponds to color class i, we write each q-coloring of G as a vector $(I_1, \ldots, I_q)$. Applying the graph container theorem (Theorem 1.6) on G, we obtain a collection of containers $\mathcal{C}$ satisfy that for every independent set I_j there is a C_ℓ such that $I_j \subseteq C_\ell$.

By the argument in Proposition 1.11, we know that each container $C \in \mathcal{C}$ has size at most

$$\frac{n}{2} + \frac{n\sqrt{2d\log n}}{2d} = \frac{n}{2} + o(n) \tag{1}$$

and the number of containers is at most

$$2^{\frac{n}{2\sqrt{2d\log n}}\cdot(\log 2d + \log\log n) + \frac{n}{\sqrt{2d\log n}}\log e} = 2^{o(n)}.$$

Therefore, the number of vectors $(C_1, \ldots, C_q)$, where $C_\ell \in \mathcal{C}$, is at most $2^{o(n)\cdot q}$. Note that for a fixed container vector $(C_1, \ldots, C_q)$, the number of colorings $(I_1, \ldots, I_q)$ such that $I_j \subseteq C_j$ is upper bounded by $\prod a_i$, where a_i is the number of containers in $\{C_1, \ldots, C_q\}$ containing the vertex $v_i \in V(G) = \{v_1, \ldots, v_n\}$.

Thus, by (1) and $a_1 + \ldots + a_n = |C_1| + \ldots + |C_q|$,

$$\text{the number of q-colorings} \leqslant \prod a_i \leqslant \left(\frac{a_1 + \ldots + a_n}{n}\right)^n \leqslant q^n \left(\frac{1}{2} + o(1)\right)^n.$$

□

References

[1] D. Galvin, *Counting Colorings of a Regular Graph*, Graphs and Combinatorics (2015) 31, 629–638.

[2] D. Galvin and P. Tetali, *On weighted graph homomorphisms*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science (2004) 63, Graphs, Morphisms and Statistical Physics, 97–104.

[3] J. Kahn, *An entropy approach to the hard-core model on bipartite graphs*, Combinatorics, Probability and Computing (2001) 10, 219–237.

[4] Y. Zhao, *The number of independent sets in a regular graph*, Combinatorics, Probability and Computing (2010) 19, 315–320.