

Expander Graphs and Their Applications (XII)

Yijia Chen
Shanghai Jiaotong University

Review of the Previous Lecture

Constraints

Definition

Let $V = \{v_1, \dots, v_n\}$ be a set of variables, and let Σ be a finite alphabet. A q -ary constraint $c = (C, i_1, \dots, i_q)$ consists of

- ▶ a q -tuple of indices $i_1, \dots, i_q \in [n]$,
- ▶ and a subset $C \subseteq \Sigma^q$ of “acceptable” values.

An assignment $\mathcal{A} : V \rightarrow \Sigma$ *satisfies* c if

$$(\mathcal{A}(v_{i_1}), \dots, \mathcal{A}(v_{i_q})) \in C.$$

The formula $X_3 \vee \neg X_7 \vee X_9$ can be viewed as a constraint $(C, 3, 7, 9)$ with

$$\{(1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1), (0, 0, 0), (0, 0, 1), (0, 1, 1)\}.$$

Constraint Satisfaction Problems (CSP)

CSP

Input: A system (i.e., set) \mathcal{C} of constraints over a set of variables V and a finite alphabet Σ .

Problem: Is there an assignment $\mathcal{A} : V \rightarrow \Sigma$ that satisfies every constraint?

3COLORABILITY as CSP

Lemma

3COLORABILITY *is reducible to* CSP.

Proof.

Let $\Sigma := \{0, 1, 2\}$. Given a graph $G = (V, E)$, we identify V with the set of variables.

For each edge $e \in E$ with end vertices v_i and v_j , we introduce a constraint

$$c_e := (C_e, i, j)$$

where

$$C_e := \{(a, b) \in \Sigma^2 \mid a \neq b\}.$$

Then

G is 3-colorable \iff there is an assignment satisfying $\{C_e \mid e \in E\}$.



Let \mathcal{C} be a set of constraints. We define

unsat(\mathcal{C}) := the smallest fraction of
unsatisfied constraints of α under any assignment

$$= \min_{\mathcal{A} \text{ an assignment}} \frac{|\{c \in \mathcal{C} \mid \mathcal{A} \text{ satisfies } c\}|}{|\mathcal{C}|}.$$

\mathcal{C} is satisfiable \iff unsat(\mathcal{C}) = 0;

\mathcal{C} is not satisfiable \iff unsat(\mathcal{C}) $\geq \frac{1}{|\mathcal{C}|}$.

Let $q \in \mathbb{N}$, $\epsilon > 0$, and Σ be a finite alphabet.

$\text{GAP-CSP}_{q,\Sigma,\epsilon}$

Input: A set \mathcal{C} of q -ary constraints over the alphabet Σ such that *either* $\text{unsat}(\mathcal{C}) = 0$ *or* $\text{unsat}(\mathcal{C}) \geq \epsilon$.

Problem: Decide whether \mathcal{C} is satisfiable.

Theorem

$\text{NP} = \text{PCP}(\log n, 1)$ if and only if for some $\epsilon > 0$ the problem $\text{GAP-CSP}_{q,\Sigma,\epsilon}$ is NP-hard (for some $q \geq 2$ and $|\Sigma| \geq 2$).

Constraint Graphs and Operations on Them

Constraint Graph

Definition

$G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ is a constraint graph, if

1. (V, E) is an undirected graph (possibly with selfloops and multi-edges), i.e., *the underlying graph* of G .
2. The set V is also viewed as a set of variables assuming values over alphabet Σ .
3. Each edge $e \in E$ carries a constraint $c(e) \subseteq \Sigma^2$ and $\mathcal{C} = \{c(e) \mid e \in E\}$. A constraint $c(e)$ is said to be satisfied by (a, b) if $(a, b) \in c(e)$.

Remark.

- ▶ *The above definition is the rephrase of a CSP with each constraint being binary.*
- ▶ *Sometimes, we also use G to refer to (V, E) .*

unsat

An assignment is a mapping $\sigma : V \rightarrow \Sigma$.

For a constraint graph $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ and an assignment σ :

$$\underline{\text{unsat}}_{\sigma}(G) := \Pr_{\substack{e \in E \text{ with} \\ \text{endvertices } u \text{ and } v}} [(\sigma(u), \sigma(v)) \notin c(e)]$$

Then

$$\underline{\text{unsat}}(G) := \min_{\sigma} \text{unsat}_{\sigma}(G)$$

We have already seen:

Theorem

Given a constraint graph $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ with $|\Sigma| = 3$, it is NP-hard to decide whether $\text{unsat}(G) = 0$.

unsat Amplification

Let $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ be a constraint graph. Then

$$G \text{ is satisfiable} \iff \text{unsat}(G) = 0$$

$$G \text{ is not satisfiable} \iff \text{unsat}(G) \geq \frac{1}{|\mathcal{C}|} = \frac{1}{|E|}.$$

Our goal: In polynomial time, compute from G another constraint graph G^* such that

$$G \text{ is satisfiable} \iff \text{unsat}(G^*) = 0$$

$$G \text{ is not satisfiable} \iff \text{unsat}(G^*) \geq \alpha.$$

Main Theorem

Definition

For every constraint graph $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$, let

$$\underline{\text{size}}(G) := |V| + |E|.$$

Theorem

There exists a Σ_0 such that the following hold.

For every finite alphabet Σ there exist two constants $C_\Sigma > 0$ and $0 < \alpha_\Sigma < 1$ and a *polynomial time algorithm* \mathbb{A}_Σ such that for every constraint graph G , the algorithm \mathbb{A}_Σ computes a constraint graph G' such that

- ▶ $\text{size}(G') \leq C_\Sigma \cdot \text{size}(G)$.
- ▶ If $\text{unsat}(G) = 0$, then $\text{unsat}(G') = 0$.
- ▶ If $\text{unsat}(G) \neq 0$, then $\text{unsat}(G') \geq \min(2 \cdot \text{unsat}(G), \alpha_\Sigma)$.

Main Theorem (cont'd)

Let Σ be a finite alphabet.

Consider the following algorithm \mathbb{G}_Σ with input a constraint graph $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$.

1. $G_0 = \langle (V_0, E_0), \Sigma_0, \mathcal{C}_0 \rangle \leftarrow \mathbb{A}_\Sigma(G)$.
2. **for** $i = 1$ **to** $\log |\mathcal{C}_0|$ **do**
3. $G_i \leftarrow \mathbb{A}_{\Sigma_0}(G)$.
4. Output $G_{\log |\mathcal{C}_0|}$.

Theorem

- ▶ $\text{unsat}(\mathbb{G}_\Sigma(G)) \geq \alpha_{\Sigma_0}$.
- ▶ For every $i \in [\log |\mathcal{C}_0|]$,

$$\text{size}(G_i) \leq C_{\Sigma_0}^i \cdot \text{size}(G_0) = C_{\Sigma_0}^{\log |\mathcal{C}_0|} \cdot \text{size}(G_0) = \text{size}(G)^{O(1)}.$$

- ▶ \mathbb{G}_Σ is a polynomial time algorithm.

Graph Powering

Fix some $d \in \mathbb{N}$.

Let $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ be a constraint graph with (V, E) being d -regular, and $t \in \mathbb{N}$.

Recall, a sequence (u_0, u_1, \dots, u_t) is a t -step walk in G if there is an edge between u_{i-1} and u_i for all $i \in [t]$.

Then we will define the following t -th power of G

$$\underline{G}^t := \langle (V, E^t), \Sigma^{d^{\lceil t/2 \rceil}}, \mathcal{C}^t \rangle$$

$$G^t := \langle (V, E^t), \Sigma^{d^{\lceil t/2 \rceil}}, \mathcal{C}^t \rangle$$

1. The vertices of G^t are the same as G . The number of edges in E^t between u and v is the number of t -step walks from u to v in G .
2. The alphabet of G^t is $\Sigma^{d^{\lceil t/2 \rceil}}$: Let

$$\Gamma(u) := \{u' \in V \mid u = u_0, u_1, \dots, u_{\lceil t/2 \rceil} = u' \text{ is a walk in } G\}.$$

Then $|\Gamma(u)| \leq d^{\lceil t/2 \rceil}$ and *by choosing some canonical order*, a value $a \in \Sigma^{d^{\lceil t/2 \rceil}}$ can be interpreted as an assignment $a : \Gamma(u) \rightarrow \Sigma$. One might think of this value as describing u 's opinion of its neighbor's values.

3. The constraint $C(e)$ associated with an edge $e \in E^t$ with end vertices u and v contains those pairs $a, b \in \Sigma^{d^{\lceil t/2 \rceil}}$ if: There is an assignment $\sigma : \Gamma(u) \cup \Gamma(v) \rightarrow \Sigma$ that satisfies every constraint $c(e) \in \mathcal{C}$ where $e \in E \cap (\Gamma(u) \times \Gamma(v))$, and such that

$$\text{for all } u' \in \Gamma(u) \text{ and } v' \in \Gamma(v), \quad \sigma(u') = a_{u'} \text{ and } \sigma(v') = b_{v'}$$

where $a_{u'}$ is the value a assigns u' , and $b_{v'}$ the value b assigns to v' .

Amplification Lemma

Clearly $\text{unsat}(G) = 0$ implies $\text{unsat}(G^t) = 0$. What about the other direction?

Lemma

Let $0 < \lambda < d$ and $|\Sigma|$ be constants. Then there exists a constant $\beta_2 = \beta_2(\lambda, d, |\Sigma|) > 0$ such that for every $t \in \mathbb{N}$ and for every d -regular constraint graph G with a selfloop on each vertex and $\lambda(G) \leq \lambda$,

$$\text{unsat}(G^t) \geq \beta_2 \cdot \sqrt{t} \cdot \min\left(\text{unsat}(G), \frac{1}{t}\right).$$

What we have lost:

- ▶ degree $d \rightarrow d^t$;
- ▶ size $|V| + |E| \rightarrow |V| + d^{t-1} \cdot |E|$.

Preprocessing Lemma

Lemma

There exist constant $0 < \lambda < d$ and $\beta_1 > 0$ such that any constraint graph G can be transferred into a constraint graph $G' := \text{prep}(G)$ such that

- ▶ G' is d -regular with selfloops and $\lambda(G') \leq \lambda < d$.
- ▶ G' has the same alphabet as G and $\text{size}(G') = O(\text{size}(G))$.
- ▶ $\beta_1 \cdot \text{unsat}(G) \leq \text{unsat}(G') \leq \text{unsat}(G)$.

Proof of the Preprocessing Lemma

Let $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ be a constraint graph.

Let $H = (V, E(H))$ be a d -degree expander with selfloops on vertex set V .

Then we let

$$G' := \langle (V', E'), \Sigma', \mathcal{C}' \rangle$$

with

- ▶ $V' := V$;
- ▶ $E' := E \dot{\cup} E(H)$;
- ▶ $\Sigma' := \Sigma$;
- ▶ $\mathcal{C}' := \mathcal{C} \dot{\cup} \bigcup_{e \in E(H)} \{(a, b) \mid a, b \in \Sigma\}$.

□

Alphabet Reduction

Lemma

There exist a constant $\beta_3 > 0$, an alphabet Σ_0 , and a linear time algorithm \mathbb{C} such that for every constraint graph $G = \langle (V, E), \Sigma, \mathcal{C} \rangle$ the algorithm \mathbb{C} computes a constraint graph G' over the alphabet Σ_0 such that

- ▶ $\text{size}(G') \leq c_\Sigma \cdot \text{size}(G)$, where c_Σ is a constant only depending on Σ .
- ▶ and $\beta_3 \cdot \text{unsat}(G) \leq \text{unsat}(G') \leq \text{unsat}(G)$.

Back to Expander Graphs

Construction of Expander Graphs

Lemma

There exist $d_0 \in \mathbb{N}$ and $h_0 > 0$ such that there is a polynomial time constructible family $\{X_n\}_{n \in \mathbb{N}}$ of d_0 -regular graphs X_n on n vertices with $h(X_n) \geq h_0$.

Using Zig-Zag product, we can construct $\{G_k\}_{k \in \mathbb{N}}$ such that:

Theorem

Every graph G_k is a $(d^{4k}, d^2, 1/2)$ -graph for all $n \in \mathbb{N}$.

Proof of the lemma. If $n = d^{4k}$ for some $k \in \mathbb{N}$, then take $X_n := G_k$ and *add appropriate number of new selfloops to each vertex.*

If $d^{4k-4} < n < d^{4k}$, then let $m := d^{4k} - n < (d^4 - 1) \cdot n$. Take G_k and

- ▶ *merge non-neighboring vertices to decrease the number of vertices to n ;*
- ▶ *add appropriate number of selfloops to every vertices.*

□