Fine Grained Complexity

5 января 2022 г.

Содержание

1. Оценки под SETH

2 сентября

1.1 Введение

Зачем нужно Fine-grained complexity? Мы отвратительны в доказательстве нижних оценок. Поэтому мы делаем следующее: берём задачу, которую очень долго не могут решить, рассматриваем её как гипотезу и в этом предположении строим условные нижние оценки на другие задачи.

Задача. (k-SAT)
Дана формула на n логических переменных в КНФ, такая что размер каждого клоза не больше k. Проверить, существует ли означивание переменных, выполняющее формулу.

Гипотеза. (ETH) [Impagliazzo and Paturi, 2001]
3-SAT не решается за время $2^{o(n)}$.

Гипотеза. (SETH) [Impagliazzo and Paturi, 2001]
Для $\forall \epsilon > 0$ найдётся $k > 0$, такое что k-SAT не решается за время $2^{(1-\epsilon)n}$.

Утверждение 1. SETH $\Rightarrow$ ETH.

Доказательство. Сведём k-SAT $\to$ 3-SAT, добавлением $(k-3)m$ новых переменных. Чтобы получить линейное разрастание числа переменных, воспользуемся Sparsification леммой (**TODO**: ссылка на лемму). $\square$

Замечание. Это единственные именно *гипотезы*, все остальные будут *conjecture*. Причина того, что эти ребята гипотезы (по словам Ивана) в том, что авторы не сильно в них верили.

Замечание. Если мы сломаем 3-SUM-conjecture, то просто получим более быстрый алгоритм для 3-SUM. Если сломаем ETH, то перевернём мир схемной сложности (**TODO**: ссылка на теорему про ETH и E^{NP}).

Определение. (Fine-grained сведение)

Будем говорить, что задача P $(\mathbf{T_1}, \mathbf{T_2})$ **fine-grained сводится** к задаче Q (пишем $P \xrightarrow{T_1,T_2} Q$), если существует такой алгоритм A, который решает P с оракульным доступом к Q так что:

- Сложность A на входе размера n составляет $\mathcal{O}(T_1(n)^{1-\alpha})$ для $\alpha > 0$
- Для $\forall \delta > 0$ найдётся $\varepsilon > 0$, так что для любого входа размера nоракульные запуски $S_1, \dots S_k$ удовлетворяют следующему условию: $\sum\limits_{i=1}^{k} T_2(S_i)^{1-\delta} \leqslant T_1(n)^{1-\varepsilon}$

1.2 Нижние оценки под SETH

Здесь и далее вместо условных нижних оценок будет писать только fg-сведения.

Задача. (ORTHOGONAL VECTORS (OV)) Даны 2 набора A и B из n векторов из $\{0,1\}^d$, где $d = o(n)$. Нужно узнать существуют ли $a \in A, b \in B$, такие что: $\sum_{i=1}^{d} a_i b_i = 0$.

Сведение. (k-SAT $\xrightarrow{2^n, n^2}$ ORTHOGONAL VECTORS) [Pătrașcu and Williams, 2010]
Построим наборы размера $2^{n/2}$ и размерности m. В первом наборе на i-ой позиции поставим 0, если данное означивание первых $n/2$ переменных выполняет i-ый клоз. Во втором аналогично. Теперь

скалярное произведение двух наборов будет равно 0 $\Leftrightarrow$ данное означивание выполняет все клозы.

Задача. (d-Hitting Set)
Дан универс $\mathcal{U}$, $|\mathcal{U}| = n$ и набор $\mathcal{S}$ подмножеств U мощности не более d. Проверить, существует ли $X \subseteq \mathcal{U}$, $|X| \leqslant k$, такой что $\forall i, S_i \cap X \neq \varnothing$.

Сведение. (k-SAT $\xrightarrow{2^n, 2^{n/2}}$ d-Hitting Set)
Возьмём в качестве универса литералы (переменные и их отрицания), в качестве множеств из $\mathcal{S}$: $\{x_i, \overline{x_i}\}$ (чтобы выбрать означивание) и $\{x_{i,1}, \dots x_{i,k}\}$ (литералы, выполняющие i-ый клоз), получаем $d \leqslant max(2, k)$.

Сведение. (k-SAT $\xrightarrow{2^n, 2^n}$ d-Hitting Set) [Cygan et al., 2016] тык
Создадим универс из n' (определим позднее) элементов, которые разобьём на группы по p, где $2 \nmid p, p \mid n'$. Заставим брать в Hitting set ровно $\lfloor p/2 \rfloor$ элементов из каждого блока: тогда каждый блок закодирует $\binom{p}{\lfloor p/2 \rfloor}$ вариантов — означивание для $\alpha_p = \lfloor \log \binom{p}{\lfloor p/2 \rfloor} \rfloor$ переменных. $\frac{\alpha_p}{p} = \frac{\lfloor \log \binom{p}{\lfloor p/2 \rfloor} \rfloor}{p} \sim \frac{\log(\frac{2^p}{\sqrt{p}})}{p} \xrightarrow[p \to \infty]{} 1$, так что размер универса будет $n' = \frac{n}{\alpha_p} p \sim n$.

Чтобы в каждом блоке бралось хотя бы по $\lfloor p/2 \rfloor$ элементов, положим все подмножества из $\lceil p/2 \rceil$ элементов в $\mathcal{S}$ (теперь, если мы взяли меньше $\lfloor p/2 \rfloor$, то дополнение этих элементов не похичено). Также докинем в $\mathcal{S}$ все дополнения подмножеств размера $\lfloor p/2 \rfloor$, которые не соответствуют означиваниям (такие могли появиться из-за округлений). Так как p — константа, всех этим множеств будет какое-то линейное от n число.

Чтобы в каждом блоке бралось не более $\lfloor p/2 \rfloor$ элементов положим $|X| = k = \frac{n}{\alpha_p} \lfloor p/2 \rfloor$.

Осталось заставить это всё выполнять клозы. Пусть клоз c_i содержит литералы $x_{i_1}, \dots, x_{i_{k_i}}$ из блоков $b_{i_1}, \dots, b_{i_{k_i}}$. Тогда переберём все означивания переменных в этих блоках, не выполняющие клоз c_i и положим объединение дополнения соответствующих им подмножеств размера $\lfloor p/2 \rfloor$ в $\mathcal{U}$. Так как p — константа, получаем линейное от m число множеств.

Можем ещё оценить d как $\max(\lceil p/2 \rceil, k \cdot \lfloor p/2 \rfloor)$

1.3 Семинар

Упражнение. Построить $(2^n, n^t)$ fg-сведение k-SAT $\rightarrow$ t-Dominating Set

Упражнение. Построить $(2^n, n^{t-1})$ fg-сведение k-SAT $\rightarrow$ t-Sparse Dominating Set ($\frac{|E(G)|}{|V(G)|} = n^{o(1)}$)

Задача. (k-SUM)
Дано k массивов $A_1, \dots, A_k$ длины n, состоящие из целых чисел из $\{-M..M\}$, где $M = n^{\mathcal{O}(1)}$. Существуют ли индексы $j_1, \dots j_k$ такие что: $\sum\limits_{i=1}^{k} A_{i,j_i} = 0$.

Упражнение. Покажите нижнюю оценку $n^{\Omega(k)}$ для k-Sum, построив цепочку fg-сведений
3-SAT $\rightarrow$ 1-in-3-SAT $\rightarrow$ k-Sum (в 1-in-3-SAT хотим выполнить клоз ровно одной переменной)

2. ETH. Нижние оценки про coloring'и

Чем хороша ETH*? Из неё вытекает оочень много условных нижних оценок на разные экспоненциальные задачи, например, на большинство задач из списка Карпа [Karp, 1972]. (Забавно, что это вообще не так для SETH: из всего большого списка только* Hitting Set *имеет условную нижнюю оценку под* SETH*). Единственная проблема* ETH *в том, что мы различаем с её помощью* $2^{o(n)}$ *и* $2^{\Theta(n)}$*, а между ними огромный зазор.*

Замечание. Для многих задач строится линейное сведение под ETH (например, число вершин в графе линейно относительно числа вершин/клозов). Для всех таких задач получаем оценку $2^{\Omega(n+m)}$.

2.1 3-Coloring

Задача. (3-Coloring) Дан граф G. Можно ли его правильно раскрасить в 3 цвета.

Задача. (Sparse 3-Coloring) Дан граф G, такой что $E(G) = \mathcal{O}(V(G)$. Можно ли его правильно раскрасить в 3 цвета.

Сведение. (3-SAT $\xrightarrow{2^{o(n)},2^{o(n+m)}}$ Sparse 3-coloring) Создадим гаджет K_3, задающий 3 цвета: T, F и N.

Создадим по 2 вершины на каждую переменную x_i и $\overline{x_i}$, соединим их с вершиной N и друг с другом: так они будут принимать различные значения из $\{T, F\}$.

Опишем OR-гаджет $(x \vee y)$: K_3, две вершины соединены с x и y соответственно, третья — выходная. Легко заметить, что она будет обязана быть покрашена в $F \Leftrightarrow x$ и y обе окрашены в F.

Переменные каждого клоза соединим двумя OR-гаджетами, выходную вершину клоза соединим с F.

Получаем 3 вершины для цветов, по 2 вершины на переменную и по 6 на каждый клоз. Рёбер тоже линейное число (поэтому получаем sparse инстанс).

Задача. (Planar 3-Coloring) Дан планарный граф G. Можно ли его правильно раскрасить в 3 цвета.

Факт. (Planar separator theorem) [Lipton and Tarjan, 1979] В планарном графе существует сбалансированный сепаратор (размер частей $\leqslant \frac{2n}{3}$) размера $\mathcal{O}(\sqrt{n})$. Более того, его можно найти за линейное время.

Алгоритм. $2^{\mathcal{O}(\sqrt{n})}$ для Planar List 3-Coloring
Найдём сепаратор, переберём его раскраску. Решим List 3-Coloring для половинок. Итого $3^{\sqrt{n}} = 2^{\sqrt{n}}$

Замечание. Часто, когда для задачи долго не получается придумать более быстрый алгоритм, оказывается, что это точная верхняя оценка. Так что сейчас мы покажем соответствующую нижнюю оценку для Planar 3-Coloring

Сведение. (Sparse 3-Coloring → Planar 3-Coloring)

Построим гаджет, исправляющий пересечения рёбер (рис 1), при раскраске в 3 цвета цвета противоположных внешних вершин совпадают (и все пары раскрасок существуют). Воткнём такой гаджет на место каждого пересечения рёбер. Так как рёбер линейно от числа вершин.

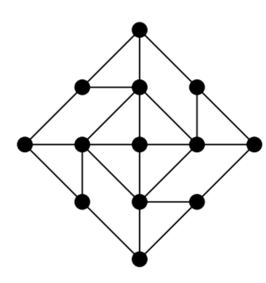

Рис. 1: Crossover gadget.

Задача. (H-coloring)

Даны два графа G и H, проверить, существует ли гомоморфизм из G в H. (гомоморфизм — отображение $g\colon G \to H$, т.ч. $\forall (u,v) \in E(G), (g(u), g(v)) \in E(H)$)

Замечание. Coloring — потому что если взять в качестве H k-клику, получится просто k-Coloring. То есть мы красим вершины G в цвета — вершины H — но не все пары цветов запрещены.

Замечание. Есть похожая задача — про миноры: проверить, что один граф является минором другого. И вот оказывается, что самым сложным минором является k-клика, её можно искать только за $k^{\Omega(k)}$.

Задача. ($(2,k)$-CSP) (Constraint satisfaction problem)
Даны n переменных, принимающих значения из алфавита размера k и множество условий (предикатов) на двух переменных (любые условия, не только ДНФ).

Сведение. (3-Coloring $\xrightarrow{2^{\Omega(n)}, 2^{\Omega(n \log k)}}$ $(2,k)$-CSP)
По графу на n вершинах построим $(2,k)$-CSP формулу на $\frac{n}{\log_3 k}$ переменных.
Каждая переменная соответствует раскраске $\log_3 k$ вершин. А условия для каждой пары переменных запрещают противоречащие раскраски. Ну и понятно, нужно ещё запретить означивания переменных, в которых вершины из одного блока покрашены неправильно.

Утверждение 2. H-Coloring — частный случай $(2,k)$-CSP.

Доказательство. Означивание переменной соответствует "цвету" вершины. И единственные предикаты, которые можем использовать, задаются матрицей смежности H.
Заметим, что из этого факта вытекает, что 3-Coloring — также частный случай $(2,k)$-CSP, в нём используется только предикат $\neq$. □

Сведение. (List H-Coloring $\xrightarrow{2^{\Omega(n \log k)}, 2^{\Omega(n \log k)}}$ H-Coloring), где $k = |V(H)|$ [Cygan et al., 2017] (скетч)
Пример: List 3-Coloring → 3-Coloring. Создадим треугольничек — "палитру" — и соединим вершины, в которых запрещены какие-то цвета с соответствующими вершинами треугольничка.
Рассмотрим такой гаджет. Его понт в том, что при любом гомоморфизме центр переходит в центр. Чтобы работало и при большом H (и не коллизилось с другими вершинами, засунем в центр клику K_{k+3}).

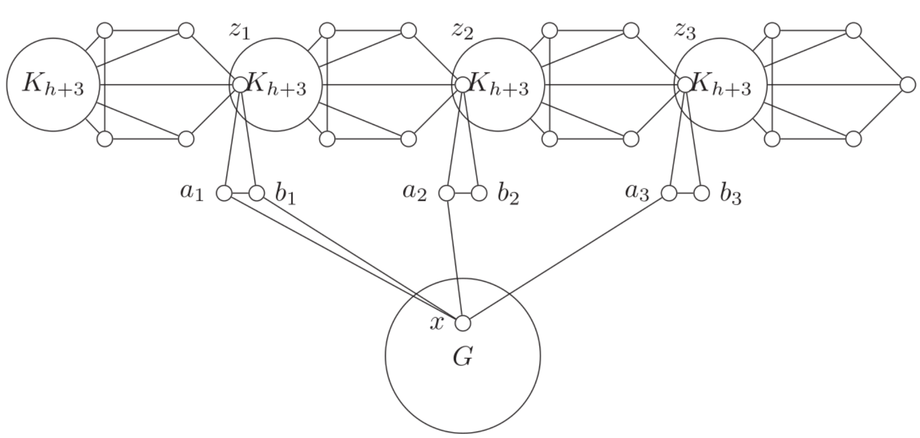

Рис. 2: Пример гаджета для $k = 3$, и вершины x, т.ч. $L(x) = \{1\}$

Теперь сцепим k таких гаджетов в цепочку. К каждому гаджету подвесим по две вершинки a_i и b_i, это наша "палитра". Теперь вершинку графа G соединим с a_i, и если её нельзя красить в i-ый цвет ещё и с b_i (рис. 2). Аналогично соединяем с "палитрой" граф H. Примерно понятно, почему это работает (но тут скетч, так что строго не будет).

Утверждение 3. [Cygan et al., 2017]

Lemma 3.2. *For any constant $d \geq 1$, there exist positive integers $\lambda = \lambda(d)$, $n_0 = n_0(d)$ and a polynomial time algorithm that for a given graph G on $n \geq n_0$ vertices of maximum degree d and a positive integer $r \leq \sqrt{\frac{n}{2\lambda}}$, finds a grouping $\tilde{G}$ of G and a coloring $\tilde{c} : V(\tilde{G}) \to [\lambda r]$ such that*

(*i*) *The number of buckets of $\tilde{G}$ is*

$$|V(\tilde{G})| \leq \frac{|V(G)|}{r};$$

(*ii*) *The coloring $\tilde{c}$ is a proper coloring of $\tilde{G}^2$;*
(*iii*) *Each bucket $B \in V(\tilde{G})$ is an independent set in G, that is, for every $u, v \in B$, $uv \notin E(G)$;*
(*iv*) *For every pair of buckets $B_1, B_2 \in V(\tilde{G})$ there is at most one edge between them in G, that is,*

$$|\{uv \in E(G) : u \in B_1, v \in B_2\}| \leq 1.$$

Сведение. (Bounded Degree 3-Coloring $\xrightarrow{2^{\Omega(n)}, 2^{\Omega(n \log k)}}$ List H-Coloring), где $k = |V(H)|$.

Хотим построить сведение примерно как для $(2,k)$-CSP — разбить вершинки на блоки, сжать блоки и решить H-Coloring для сжатого графа. Проблема в том, что мы не можем прослеживать связи между вершинами из разных блоков. Поэтому сведение будет более хитрым и потребует "фактов из теории графов".

Возьмём разбиение из утв. 3 и его раскраску (будет говорить, что она задаёт не цвета, а лейблы, цвета — в исходном 3-Coloring'е). Блоки образуют независимые множества и между двумя блоками не более одного ребра. Более того, так как раскраска переносится на $\widetilde{G}^2$, то для каждого блока все его соседи разных лейблов.

Для блока зададим 4-ичный вектор длины $\chi(\widetilde{G})$. На t-ой позиции этого вектора запишем цвет (в 3-Coloring'е) соседа этого блока с лейблом t (или 0, если такого нет). Тогда вершина в H задаёт лейбл блока и кодировку. List H-Coloring'ом (частью про List) добиваемся того, что блок отображается только в свой лейбл. Соединяем в H вершины, раскраски которых согласованы ($\phi_a(b) \neq \phi_b(a) \vee \phi_a(b)\phi_b(a) = 0$).

Теперь оценим время. Выберем размер блока равным r, тогда $|V(\widetilde{G})| \leqslant \frac{n}{r}$, $|V(H)| \leqslant \chi(\widetilde{G}) 4^{\widetilde{G}} = \mathcal{O}(r 4^{\mathcal{O}(r)})$. Выбрав $r = \mathcal{O}(\log n)$ получаем $|V(G)| = \mathcal{O}(\frac{n}{\log n})$, $|V(H)| = n^{\mathcal{O}(1)}$.

Задача. (Subgraph Isomorphism) Даны графы G и H, проверить, является ли H подграфом G.

Сведение. (H-Coloring $\xrightarrow{2^{\Omega(n \log n)}, 2^{\Omega(n \log n)}}$ Subgraph Isomorphism)

Переберём a_i, $\sum\limits_{i=1}^{|V(H)|} a_i = |V(G)|$ — сколько вершин G отобразится в i-ую вершину H. Продублируем i-ую вершину H i раз, поищем изоморфизм из H в G. Так как неупорядоченных разбиений на слагаемые будет не более $2^{|V(G)|} = 2^n$ получили нужную оценку.

2.2 Семинар

Упражнение. $\forall \alpha > 0$ построить $(2^n, n^{1+\alpha})$ fg-сведение Ham Cycle $\to$ 3-Sum

Задача. (Max Inner Product)
Даны 2 набора A и B из n векторов из $\{0,1\}^d$, где $d = o(n)$. Найти $\max\{\langle a, b\rangle | a \in A, b \in B\}$.

Упражнение. Покажите, что Max Inner Product не решается за время $\mathcal{O}(n^{2-\varepsilon})$ под SETH.

Упражнение. Покажите оценку в $2^{\Omega(n)}$ на Bounded Degree 3-Coloring ($\frac{|E(G)|}{|V(G)|} = n^{o(1)}$) под ETH.

Задача. (Cross Matching) Дан граф G с разбиением вершин на две доли по n вершин. Проверить, существует ли такое совершенное паросочетание M в G, что концы каждого ребра паросочетания находятся в разных долях и G/M (стянутые рёбра) образует клику.

Упражнение. [Fomin et al., 2021] Покажите оценку в $n^{\Omega(n)}$ на задачу Cross Matching под ETH.

Список литературы

[Cygan et al., 2016] Cygan, M., Dell, H., Lokshtanov, D., Marx, D., Nederlof, J., Okamoto, Y., Paturi, R., Saurabh, S., and Wahlström, M. (2016). On problems as hard as cnf-sat. *ACM Trans. Algorithms*, 12(3).

[Cygan et al., 2017] Cygan, M., Fomin, F. V., Golovnev, A., Kulikov, A. S., Mihajlin, I., Pachocki, J., and Socała, A. (2017). Tight lower bounds on graph embedding problems. *J. ACM*, 64(3).

[Fomin et al., 2021] Fomin, F. V., Lokshtanov, D., Mihajlin, I., Saurabh, S., and Zehavi, M. (2021). Computation of hadwiger number and related contraction problems: Tight lower bounds. *ACM Trans. Comput. Theory*, 13(2).

[Impagliazzo and Paturi, 2001] Impagliazzo, R. and Paturi, R. (2001). On the complexity of k-sat. *Journal of Computer and System Sciences*, 62(2):367–375.

[Karp, 1972] Karp, R. M. (1972). *Reducibility among Combinatorial Problems*, pages 85–103. Springer US, Boston, MA.

[Lipton and Tarjan, 1979] Lipton, R. J. and Tarjan, R. E. (1979). A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189.

[Pătrașcu and Williams, 2010] Pătrașcu, M. and Williams, R. (2010). *On the possibility of faster SAT algorithms*, pages 1065–1075.