Лекция №4.

Алгоритм перечисления для нахождения кратчайшего вектора в решётке. BKZ-редукция базиса.

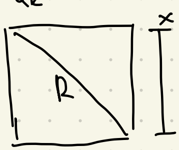
ОПР 1 Задача нахождения кратчайшего вектора (SVP shortest vector problem): в заданной решётке L, найти $v \in L \setminus \{0\}$, т.ч. $\|v\| = \lambda_1(L)$

B - базис, найти $v = Bx$, $\underset{\neq 0}{x \in \mathbb{Z}^n}$, т.ч. $\|v\|$ - min. ненулевой.

I Enumeration Algm (алг-м перечисления) [Kannan'88, Finke-Post'83]

Находит кратчайший ненулевой вектор в решётке $L(B)$, $B \in \mathbb{Z}^{n \times n}$, используя R-фактор. Идея: перечислить все $x \in \mathbb{Z}^n$ $\underset{(x_1, x_2, \dots, x_n)}{=}$: $\|Bx\| < K$ ($K \in \mathbb{R}$ - граница).

$$\|\underset{QR}{B}x\|^2 = \|Rx\|^2 = \left\| \left(\sum_{i=1}^{n} r_{1i} x_i, \ \sum_{i=2}^{n} r_{2i} x_i, \ \dots, \ r_{nn} x_n \right) \right\|^2 = \sum_{j=1}^{n} \left(\sum_{i \geq j}^{n} r_{ji} x_i \right)^2 \quad (1)$$



① Если $\|Bx\|^2 < K^2 \Rightarrow (r_{nn} x_n)^2 < K^2$

Т.к. $x_n \in \mathbb{Z}^n$, то $|x_n| < \frac{K}{r_{nn}}$.

Всего имеем $\left(2 \cdot \frac{K}{r_{nn}} + 1\right)$ значений для x_n.

② Для фиксированного x_n, ∗ 2 последних слагаемых в (1):

фиксирована

$$\left(r_{n-1,n-1} x_{n-1} + r_{n-1,n} \boxed{x_n}\right)^2 + \left(r_{n,n} \cdot \boxed{x_n}\right)^2 < K$$

$$\Updownarrow$$

$$\left| x_{n-1} + \frac{r_{n-1,n}}{r_{n-1,n-1}} \cdot x_n \right| \leq \left(\frac{K - r_{n,n} \cdot x_n}{r_{n-1,n-1}} \right)^{1/2}$$

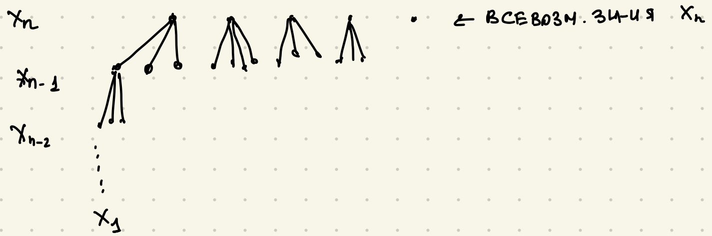
Для фиксированного x_n, перечисляем $x_{n-1} \in \mathbb{Z}$, удовлетворяющие

Всего для x_{n-1} (при фикс. x_n) имеем $\leq \left(\frac{2K}{r_{n-1,n-1}} + 1 \right)$ значений.

Аналогично, для x_i (при фиксированных $x_{i+1}\ldots x_n$) всего имеем $\left(\frac{2K}{r_{ii}}+1\right)$ подходящих значений.

Продвигаясь до x_1, получаем все вектора в решётке длины, меньше K, выбираем из них минимальный.

Реализация такого алгоритма – проход по дереву (в глубину / depth-first для оптимизации памяти).



Время работы алгоритма перечисления

Лемма 1 (св-ва LLL редуцированного базиса). Пусть $\delta \in (\frac{1}{2}, 1)$ и положим $\alpha := \frac{1}{\sqrt{\delta - 1/4}}$. Для B – LLL редуцированного базиса с пар-ом δ справедливо:

1) $\|b_1\| \le \alpha^{n-1} \lambda_1(L)$
2) $\|b_1\| \le \alpha^{\frac{n-1}{2}} (\det L)^{\frac{1}{n}}$
3) $\frac{r_{ii}}{r_{i+1,i+1}} \le \alpha \quad \forall i < n.$

◁ Док-во на практике ▷

Время работы алг-ма перечисления определено $\underbrace{|\text{дерево перечислений}|}_{\sum_{j=1}^{n} \prod_{i \le j} \left(\frac{2K}{r_{ii}}+1\right)} \cdot poly(n)$

Оценим $\sum_{j=1}^{n} \prod_{i \le j} \left(\frac{2K}{r_{ii}}+1\right)$.

По Лемме 1, $\frac{r_{11}}{r_{ii}} \le \alpha^{i-1}$, и мы можем взять в качестве границы K - длину 1ого вектора LLL редуцированного базиса: $K = r_{11} = \|b_1\| \le \alpha^{\frac{n-1}{2}} (\det L)^{\frac{1}{n}}$

$$\sum_{j=1}^{n} \prod_{i\le j}\left(\frac{2K}{r_{ii}}+1\right) = \sum_{j=1}^{n} \prod_{i\le j}\left(\underbrace{\frac{2r_{11}}{r_{ii}}}_{2d^{i-1}}+1\right) = \sum_{j=1}^{n}\prod_{i\le j}\left(2^{i}+1\right) \le \sum_{j=1}^{n} \prod_{i\le j} 3^{i} \le$$

берём $d=2$ $\left(\delta=\frac{1}{2}\right)$

$$\le n\cdot \prod_{i=1}^{n} 3^{i} = \underbrace{n\cdot 3^{n^2}}_{\downarrow \; 2^{O(n^2)}}$$ – двойная экспоненциальная сложность.

<u>Суть</u>: чем медленнее убывают r_{ii} (т.е. чем меньше граница на $\frac{r_{11}}{r_{ii}}$), тем "уже"/меньше становится дерево перечисления $\Rightarrow$ тем быстрее работает алгоритм.

$\exists$ "предобработка" исходного базиса, т.ч. последние r_{ii} становятся большими. Это позволяет уменьшить время работы алг-ма до

$$\le n^{\frac{1}{2e}\cdot n + o(n)} = \underbrace{2^{\frac{1}{2e} n\cdot \lg n + o(n \lg n)}}_{\text{суперэкспоненциальная}}$$ – время

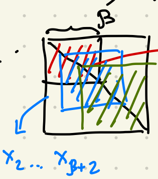
Память: $poly(n)$

II BKZ-редукция (block Korkin–Zolotarev)

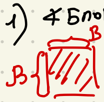
(Шнорр / Schnorr '87)
(Шнорр-Ойхнер / Schnorr - Euchner '94)

LLL: блок р-ти 2 $\Rightarrow$ BKZ: блок р-ти $\beta \in [2, n]$ ← основной пар-р алгоритма

$B = QR = Q \cdot$



1) $\forall$ блок $\beta\times\beta$, вырезав его из R-фактора



– R-фактор решётки – это "проективная" решётка р-ти β.

2) Вызываем SVP (алг-м перечисления) на этом R-факторе. $\Rightarrow$ кратчайший вектор в решётке р-ти β.

Время работы: $2^{O(\beta \log \beta)}$ или $2^{O(\beta)}$

3) "Добавляем" этот кратчайший вектор в базис B (алг. Бабая, см. будущие лекции)

4) Запускаем LLL $\left[B \,\middle|\, \begin{matrix}\text{кратч.}\\ \text{вектор}\end{matrix}\right] \Rightarrow$ убираем лин. зависимость

5) Повторяем процедуру для

$$R[(i+1), (i+1)+\beta] \times [i+1, (i+1)+\beta]$$

6) Повторяем шаги 1)–5) $poly(n)$ раз.

Лемма 2 (качество BKZ редукции)

BKZ-алгоритм, запущенный на решётке L с параметром β, возвращает BKZ-редуцированный базис с вектором b_1, удовлетворяющим

$$\|b_1\| \le \beta^{\frac{n-1}{\beta-1}} \lambda_1(L).$$

◁ Док-во на практике ▷.

Время работы BKZ: доминирует шаг 2) (SVP) $2^{O(\beta \lg \beta)}$ или $2^{O(\beta)}$ (если ограничить число "туров" функцией poly(n). Здесь "тур" — проход блоков длины β от начала базиса до конца).