Lattice Seminar

Babai Nearest Plane Algorithm

Brian Shaft

5/06/19

Lattice Seminar

Babai Nearest Plane Algorithm

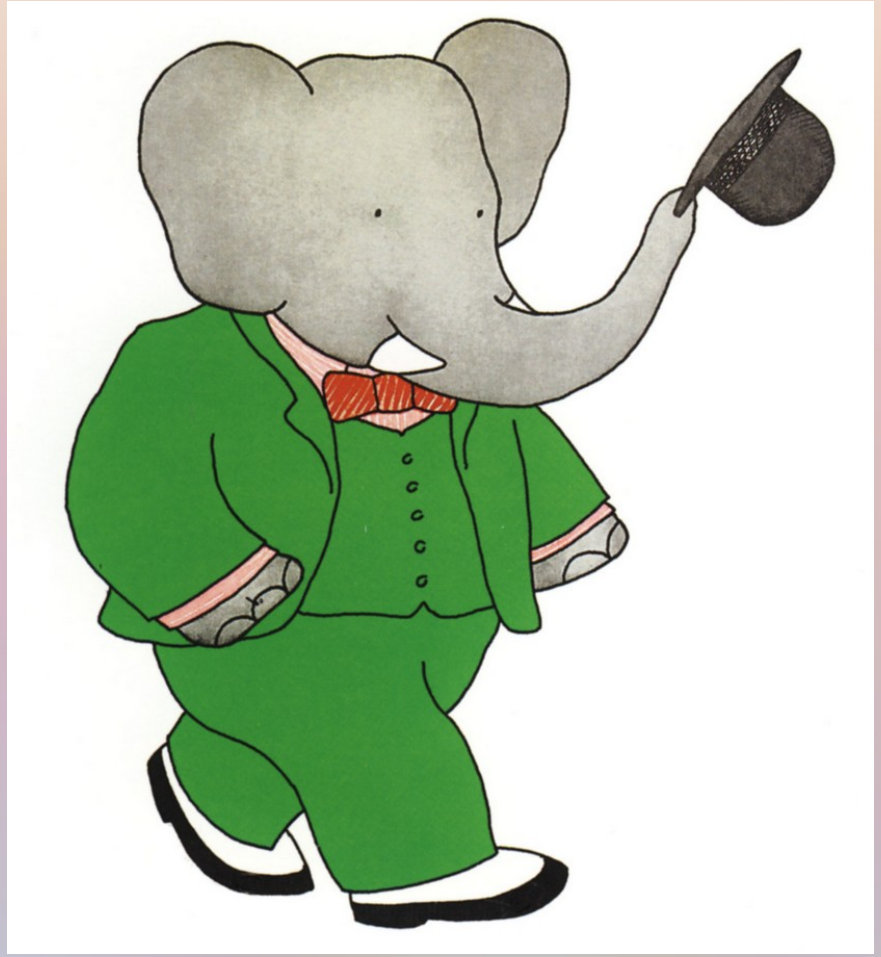
Brian Shaft

5/06/19

Introduction

- Babai
- Observations on LLL and Euclid
- Babai SVP Approximation Algorithms
  - Rounding
  - Nearest Plane

Babar nee 1931

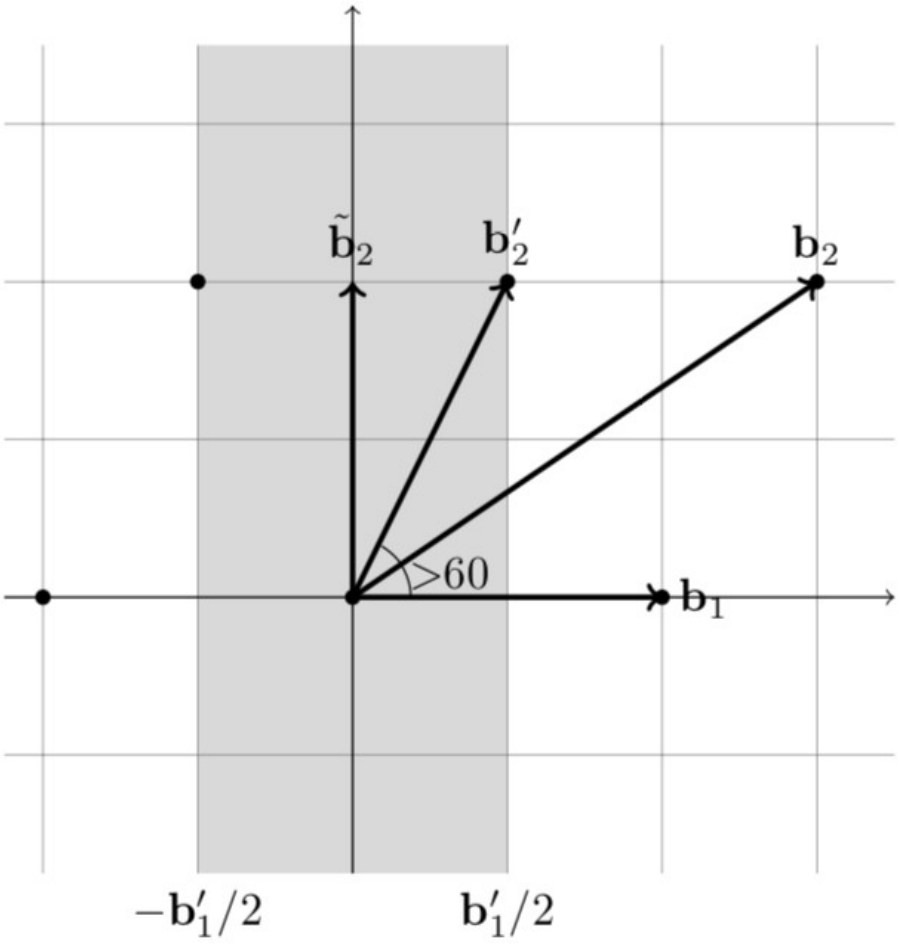


Babai nee 1951

Babai Laszlo

- Colleague of Lovasz at Eötvös Loránd University
- Thence to U Chicago (Lovasz to Yale)
- Over 180 Papers
- Still Very Active
- Gödel/Knuth/Dijkstra ... Prizes

LLL Reduction Step



The LLL-reduced basis of $[\mathbf{b}_1 = (2,0), \mathbf{b}_2 = (3,2)]$ is $[\mathbf{b}'_1 = (2,0), \mathbf{b}'_2 = (1,2)]$

Geometric Euclid Video

- https://www.youtube.com/watch?v=kiFfp-HAu64
- Euclid Algorithm starts at 6:16
- Burroughs B6700 Console at 14:17

CVP – Closest Vector Problem

For a lattice basis $L \subset \mathbb{R}^n$ and

a target vector $t \in \mathbb{R}^n$

and an approximation factor $\gamma = \gamma(n) \geq 1$

Find a lattice vector $x \in L$ where

$$\|x - t\| \leq \gamma \cdot \mathrm{dist}(t, L)$$

CVP – Closest Vector Problem

- Search
- Optimization
- Decisional

CVP – Search

For a lattice basis $L \subset \mathbb{R}^n$ and

a target vector $t \in \mathbb{R}^n$

and an approximation factor $\gamma = \gamma(n) \geq 1$

Find a lattice vector $x \in L$ where

$$\|x - t\| \leq \gamma \cdot \mathrm{dist}(t, L)$$

CVP – Search

For a lattice basis $L \subset \mathbb{R}^n$ and

a target vector $t \in \mathbb{R}^n$

and an approximation factor $\gamma = \gamma(n) \geq 1$

Find a lattice vector $x \in L$ where

$\|x-t\| \leq \gamma \cdot \mathrm{dist}(t,L)$

SVP can be reduced to CVP

CVP – Search

For a lattice basis $L \subset R^n$ and

a target vector $t \in R^n$

and an approximation factor $\gamma = \gamma(n) \geq 1$

Find a lattice vector $x \in L$ where

$\|x-t\| \leq \gamma \cdot dist(t,L)$

SVP can be reduced to CVP

NP-Complete for $\gamma \leq n^{c/\log\log n}$

Babai's Algorithms

- Use Search Approach
- If basis LLL-reduced the output has tighter bounds to the true CVP.

Rounding Algorithm

- Advantages
  - Simpler (No Gram-Schmidt stage)
  - Faster
- Negatives
  - Not as reliable as Nearest Plane Algorithm

Rounding Algorithm

Let $\underline{b}_1,\ldots,\underline{b}_n$ be a basis for a full rank lattice in $\mathbb{R}^n$. Given a target $\underline{w} \in \mathbb{R}^n$ we can write:

$$\underline{w} = \sum_{i=1}^{n} l_i \underline{b}_i$$

Rounding Algorithm

Just solve the coefficients (l_i) then integerize to get $\underline{v}$

$$\underline{v} = \sum_{i=1}^{n} \lfloor l_i \rceil \underline{b}_i$$

Babai proved that for an LLL basis and any $\underline{u}$ in L:

$$\|\underline{w} - \underline{v}\| \leq (1 + 2n(9/2)^{n/2})\|\underline{w} - \underline{u}\|$$

Nearest (Hyper)Plane Algorithm

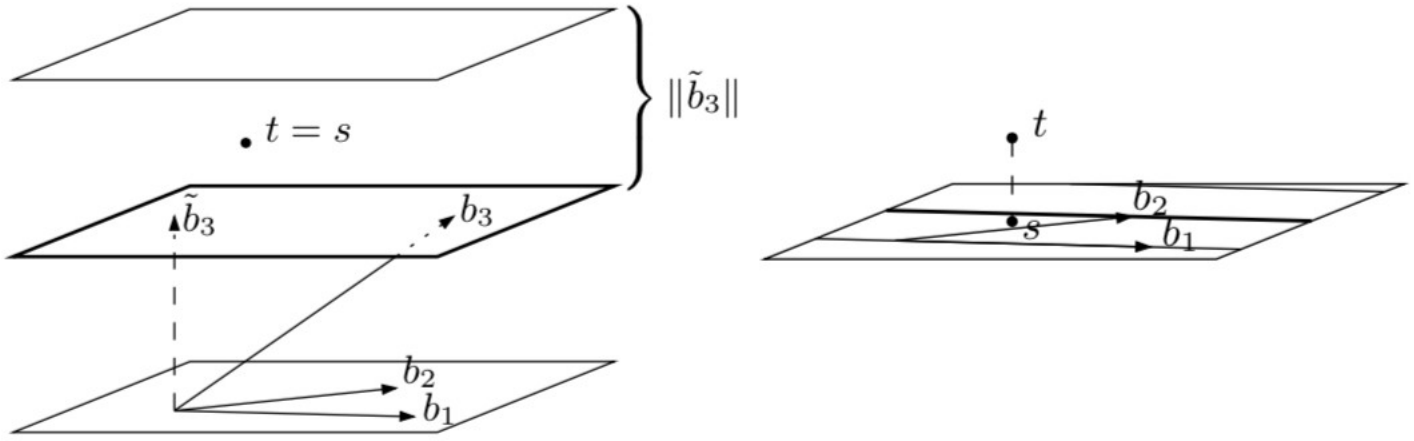
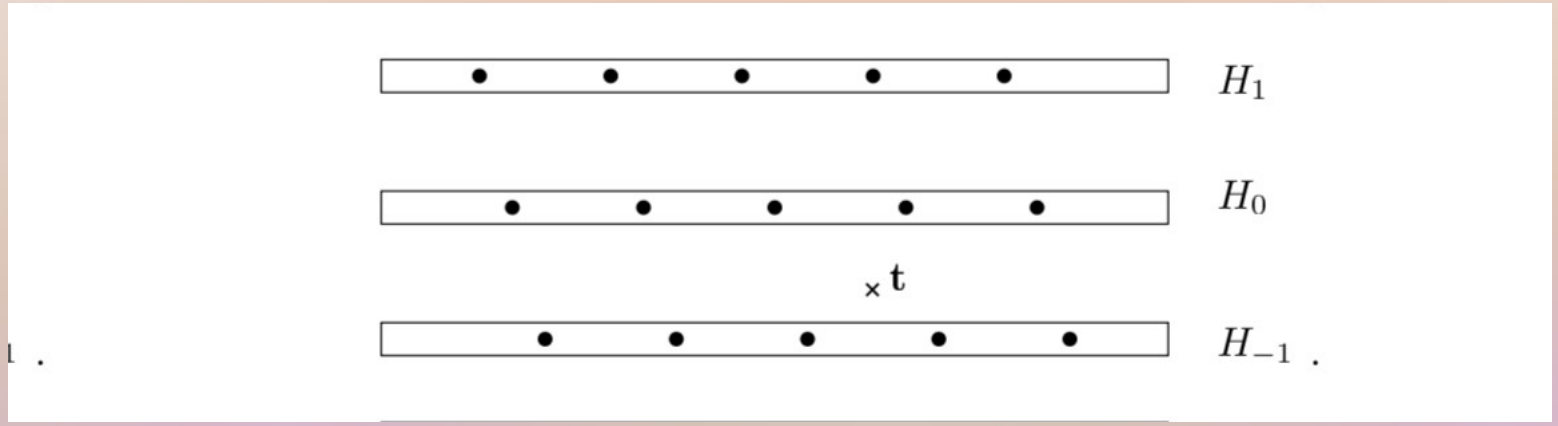


Figure 1: The nearest plane algorithm for a rank 3 lattice and the resulting rank 2 instance. The chosen hyperplanes are thicker.

1. Let s be the projection of t on $\mathrm{span}(b_1, \ldots, b_n)$.
2. Find c such that the hyperplane $c\tilde{b}_n + \mathrm{span}(b_1, \ldots, b_{n-1})$ is as close as possible to s.
3. Let $s' = s - cb_n$. Call recursively with s' and $\mathcal{L}(b_1, \ldots, b_{n-1})$. Let x' be the answer.
4. Return $x = x' + cb_n$.

Nearest (Hyper)Plane Algorithm

Or in 2 dimensions



Algorithm finds vector in H_{-1}, but closest is in H_0

Going to 2 closest planes would fix this case

Babai Recursive

- Curse + Curse

$$
\begin{aligned}
&\texttt{NearestPlane}(\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_n], \mathbf{t}):\\
&\texttt{if } n = 0 \texttt{ then return } \mathbf{0}\\
&\texttt{else } \mathbf{B}^* \leftarrow \texttt{GramSchmidt}(\mathbf{B})\\
&\qquad c \leftarrow \left\lfloor \frac{\langle \mathbf{t}, \mathbf{b}_n^* \rangle}{\|\mathbf{b}_n^*\|^2} \right\rceil\\
&\qquad \texttt{return } c\mathbf{b}_n + \texttt{NearestPlane}([\mathbf{b}_1, \ldots, \mathbf{b}_{n-1}], \mathbf{t} - c\mathbf{b}_n)
\end{aligned}
$$

Babai Iterative

Compute Gram-Schmidt basis $\underline{b}_1^*, \ldots, \underline{b}_n^*$
Set $\underline{w}_n = \underline{w}$
for $i = n$ downto 1 **do**
    Compute $l_i = \langle \underline{w}_i, \underline{b}_i^* \rangle / \langle \underline{b}_i^*, \underline{b}_i^* \rangle$
    Set $\underline{y}_i = \lfloor l_i \rceil \underline{b}_i$
    Set $\underline{w}_{i-1} = \underline{w}_i - (l_i - \lfloor l_i \rceil)\underline{b}_i^* - \lfloor l_i \rceil \underline{b}_i$
end for
return $\underline{v} = \underline{y}_1 + \cdots + \underline{y}_n$

Approximation Factor

Let $\{b_1, \ldots, b_n\}$ be LLL reduced (with respect to the Euclidean norm, and with factor $\delta = 3/4$). If $\underline{v}$ is the output of Babai's nearest plane algorithm on input $\underline{w}$ then

$$\|\underline{w} - \underline{v}\|^2 \leq \frac{2^n - 1}{4} \|\underline{b}_n^*\|^2.$$

See Galbraith Ch.18 P.385