Multi-Base Representations and their Minimal Hamming Weight

Daniel Krenn

(joint work in progress with *Vorapong Suppakitpaisarn* and *Stephan Wagner*)

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT | WIEN GRAZ

May 24, 2018

FWF Der Wissenschaftsfonds.

Supported by the Austrian Science Fund (FWF), project P28466.

Multi-Base Representations

Representations

$$n = \sum_j d_j p_1^{\alpha_{1j}} p_2^{\alpha_{2j}} \dots p_m^{\alpha_{mj}}$$

- digits d_j out of digit set $\{0, 1, \dots, d-1\}$
- bases $p_1, \dots, p_m$
  (multiplicatively independent integers ≥ 2)
- nonnegative integers α_{ij}
- all power-products $p_1^{\alpha_{1j}} p_2^{\alpha_{2j}} \dots p_m^{\alpha_{mj}}$ distinct

Multi-Base Representations

Representations

$$n = \sum_j d_j p_1^{\alpha_{1j}} p_2^{\alpha_{2j}} \dots p_m^{\alpha_{mj}}$$

- digits d_j out of digit set $\{0, 1, \dots, d-1\}$
- bases $p_1, \dots, p_m$
  (multiplicatively independent integers ≥ 2)
- nonnegative integers α_{ij}
- all power-products $p_1^{\alpha_{1j}} p_2^{\alpha_{2j}} \dots p_m^{\alpha_{mj}}$ distinct

?

Question

How good is the "best possible"
(minimal Hamming weight)
representation of a number?

Motivation from Cryptography

calculate

$$nP = P + \cdots + P$$

as efficiently as possible

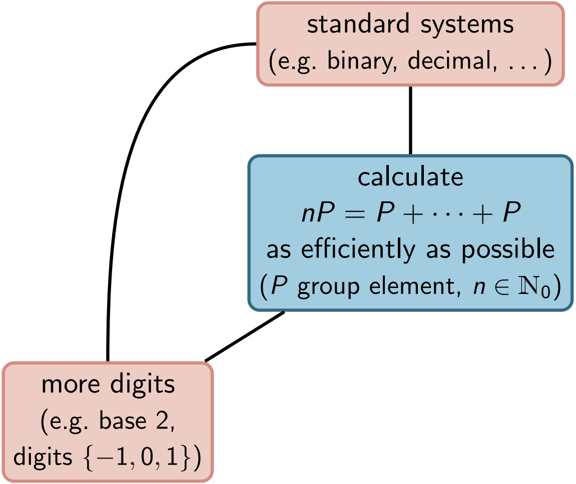
(P group element, $n \in \mathbb{N}_0$)

Motivation from Cryptography

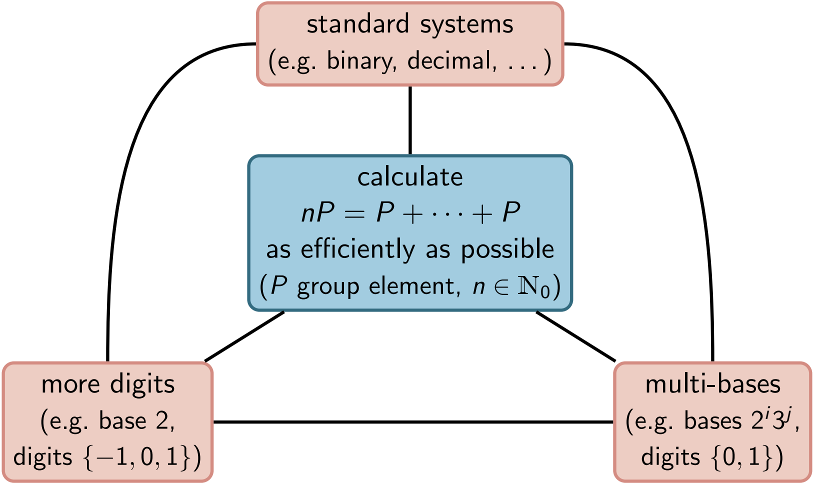
standard systems
(e.g. binary, decimal, ...)

calculate
$$nP = P + \cdots + P$$
as efficiently as possible
(P group element, $n \in \mathbb{N}_0$)

Motivation from Cryptography



Motivation from Cryptography



Hamming Weight

- multi-base representations

$$n = \sum_j d_j p_1^{\alpha_{1j}} p_2^{\alpha_{2j}} \dots p_m^{\alpha_{mj}}$$

  - digits d_j out of digit set $\{0, 1, \dots, d-1\}$
  - bases $p_1, \dots, p_m$ (multiplicatively independent integers ≥ 2)

Hamming Weight

number of $d_j \neq 0$ in representation

Hamming Weight

- multi-base representations

$$n = \sum_j d_j p_1^{\alpha_{1j}} p_2^{\alpha_{2j}} \dots p_m^{\alpha_{mj}}$$

  - digits d_j out of digit set $\{0, 1, \dots, d-1\}$
  - bases $p_1, \dots, p_m$ (multiplicatively independent integers ≥ 2)

Hamming Weight

number of $d_j \neq 0$ in representation

Minimal Hamming Weight

minimal among all multi-base representations of n with the same bases and digit set

- "measures" efficiency of representation

Minimal Hamming Weight

Theorem (K–Suppakitpaisarn–Wagner 2018)

- *fix bases* $p_1, \ldots, p_m$ $(m \geq 2)$ *multiplicatively independent*
- *fix digit set containing* 1
- *there exist positive constants* K_1 *and* K_2
  - (U) *each integers* n *has representation with Hamming weight at most* $K_1 \frac{\log n}{\log \log n}$
  - (L) *infinitely many positive integers* n *with no representation with Hamming weight less than* $K_2 \frac{\log n}{\log \log n}$

Some Properties

- smallest numbers with given weight (bases 2, 3)

weight	1	2	3	4	5	6	7	...
number	1	5	23	431	18431	3448733	1441896119	...

(sequence A018899 in the OEIS)

- finding minimal expansion seems to be hard
- compute approximation $2^i 3^j \leq n$
  (Berthé–Imbert 2009)

Number of Representations

Theorem (K–Ralaivaosaona–Wagner 2014)

- *fix bases* $p_1, \ldots, p_m$ $(m \geq 2)$
- *fix digit set* $\{0, \ldots, d-1\}$
- *number of multi-base representations* P_n *of* n

$$\begin{aligned}\log P_n = \kappa(\log n)^m &+ C_1(\log n)^{m-1}\log\log n\\ &+ C_2(\log n)^{m-1}\\ &+ O\big((\log n)^{m-2}\log\log n\big)\end{aligned}$$

- *with*

$$\kappa = \frac{\log d}{m!}\prod_{i=1}^{m}\frac{1}{\log p_i}$$

Distribution of the Hamming Weight

Theorem (K–Ralaivaosaona–Wagner 2014)

- *fix bases* $p_1, \ldots, p_m$ $(m \geq 2)$
- *fix digit set* $\{0, \ldots, d-1\}$
- *Hamming weight of uniformly random multi-base representation of* n*:*
  - *Gaussian/normal distribution (as* $n \to \infty$*)*
  - *expectation*

$$\mu_n = \frac{\kappa(d-1)}{d \log d} (\log n)^m + \mathcal{O}((\log n)^{m-1} \log \log n)$$

  - *variance*

$$\sigma_n^2 = \frac{\kappa(d-1)}{d^2 \log d} (\log n)^m + \mathcal{O}((\log n)^{m-1} \log \log n)$$

Single-base Representations

p

$$n = \sum_j d_j p^{\alpha_j}$$

- digits d_j out of finite digit set
- integer base $p \geq 2$

- Hamming weight
  - average order of magnitude is $\log n$
  - worst case (maximum) also $\log n$
- minimal Hamming weight
  - number of minimal representations *(Grabner–Heuberger 2006)*
  - compute minimal expansion *(Phillips–Burgess 2004, Heuberger–Muir 2009)*

Greedy Algorithm

Natural Greedy Algorithm

- input integer n
- add largest power-product $p_1^{\alpha_1} \dots p_m^{\alpha_m}$ less or equal to n
- continue with $n - p_1^{\alpha_1} \dots p_m^{\alpha_m}$
- output

$$n = \sum_j d_j p_1^{\alpha_{1j}} p_2^{\alpha_{2j}} \dots p_m^{\alpha_{mj}}$$

Greedy Algorithm

Natural Greedy Algorithm

- input integer n
- add largest power-product $p_1^{\alpha_1} \dots p_m^{\alpha_m}$ less or equal to n
- continue with $n - p_1^{\alpha_1} \dots p_m^{\alpha_m}$
- output

$$n = \sum_j d_j p_1^{\alpha_{1j}} p_2^{\alpha_{2j}} \dots p_m^{\alpha_{mj}}$$

- Greedy algorithm $\not\Rightarrow$ minimal representation
- smallest counter-example

$$41 = 2^2 3^2 + 2^2 + 1 = 2^5 + 3^2$$

History & Related Work

Upper Bound

natural greedy algorithm with input n terminates after

$$\mathcal{O}\Big(\frac{\log n}{\log\log n}\Big)$$

steps

- bases 2, 3, generalizes to arbitrary multi-base of primes *(Dimitrov–Jullien–Miller 1998)*
  $\rightsquigarrow$ “On the maximal distance between integers composed of small primes” *(Tijdeman 1974)*

History & Related Work

Upper Bound

natural greedy algorithm with input n terminates after

$$\mathcal{O}\left(\frac{\log n}{\log \log n}\right)$$

steps

- bases 2, 3, generalizes to arbitrary multi-base of primes *(Dimitrov–Jullien–Miller 1998)*
  $\rightsquigarrow$ “On the maximal distance between integers composed of small primes” *(Tijdeman 1974)*
- bases 2, 3, 5 *(Yu–Wang–Li–Tian 2013)*
- sharpness of bound for double-base expansions *(Chalermsook–Imai–Suppakitpaisarn 2015)*

Termination of Greedy Algorithm

Corollary (K–Suppakitpaisarn–Wagner 2018)

- *fix bases* $p_1, \ldots, p_m$ $(m \geq 2)$ *multiplicatively independent*
- *natural greedy algorithm with input* n *terminates after* $\mathcal{O}\left(\frac{\log n}{\log \log n}\right)$ *steps*
- *bound is sharp*
- *output contains only digits* 0 *and* 1

Proof of Upper Bound

Approximation by Power-products

there are positive constants C and κ with

$$ne^{-C(\log n)^{-\kappa}} \le p_1^{\alpha_1} \dots p_m^{\alpha_m} \le n$$

Proof of Upper Bound

Approximation by Power-products

there are positive constants C and κ with

$$n e^{-C(\log n)^{-\kappa}} \leq p_1^{\alpha_1} \dots p_m^{\alpha_m} \leq n$$

- two bases p and q multiplicatively independent
- set $\lambda = \log_p q$ and $M = \lceil \log_q n \rceil$

Proof of Upper Bound

Approximation by Power-products

there are positive constants C and κ with

$$ne^{-C(\log n)^{-\kappa}} \leq p_1^{\alpha_1} \dots p_m^{\alpha_m} \leq n$$

- two bases p and q multiplicatively independent
- set $\lambda = \log_p q$ and $M = \lceil \log_q n \rceil$
- discrepancy of sequence $(\{\lambda m\})_{m=0}^{M-1}$ is $\leq C_1 M^{-\kappa}$
- discrepancy bounds largest gap in sequence
- $$\{\log_p n\} - C_1 M^{-\kappa} \leq \{\lambda m\} \leq \{\log_p n\}$$

$\approx$

Proof of Upper Bound

Approximation by Power-products

there are positive constants C and κ with

$$ne^{-C(\log n)^{-\kappa}} \leq p_1^{\alpha_1} \dots p_m^{\alpha_m} \leq n$$

- two bases p and q multiplicatively independent
- set $\lambda = \log_p q$ and $M = \lceil \log_q n \rceil$
- discrepancy of sequence $(\{\lambda m\})_{m=0}^{M-1}$ is $\leq C_1 M^{-\kappa}$
- discrepancy bounds largest gap in sequence
- $$\begin{aligned} \{\log_p n\} - C_1 M^{-\kappa} &\leq \{\lambda m\} \leq \{\log_p n\} \\ \log_p n - C_1 M^{-\kappa} &\leq \ell + \lambda m \leq \log_p n \\ ne^{-C(\log n)^{-\kappa}} &\leq p^\ell q^m \leq n \end{aligned}$$

Proof of Upper Bound

Approximation by Power-products

there are positive constants C and κ with

$$ne^{-C(\log n)^{-\kappa}} \le p_1^{\alpha_1} \dots p_m^{\alpha_m} \le n$$

- two bases p and q multiplicatively independent
- set $\lambda = \log_p q$ and $M = \lceil \log_q n \rceil$
- discrepancy of sequence $(\{\lambda m\})_{m=0}^{M-1}$ is $\le C_1 M^{-\kappa}$
- discrepancy bounds largest gap in sequence
- $$\begin{aligned} \{\log_p n\} - C_1 M^{-\kappa} &\le \{\lambda m\} \le \{\log_p n\} \\ \log_p n - C_1 M^{-\kappa} &\le \ell + \lambda m \le \log_p n \\ ne^{-C(\log n)^{-\kappa}} &\le p^\ell q^m \le n \end{aligned}$$

$\Rightarrow$ upper bound follows

History & Related Work

"Lower" Bound / Sharpness

infinitely many integers n whose minimal Hamming weight is greater than

$$K_2 \frac{\log n}{\log \log n \cdot \log \log \log n}$$

- bases 2, 3 *(Dimitrov–Howe 2011)*
- bases 2, 3, 5 *(Yu–Wang–Li–Tian 2013)*

History & Related Work

“Lower” Bound / Sharpness

infinitely many integers n whose minimal Hamming weight is greater than

$$K_2 \frac{\log n}{\log \log n \cdot \cancel{\log \log \log n}}$$

- bases 2, 3 *(Dimitrov–Howe 2011)*
- bases 2, 3, 5 *(Yu–Wang–Li–Tian 2013)*

Proof of Sharpness: Counting Representations

- number of different power-products
  appearing in multi-base representations of $\{1, 2, \ldots, N\}$

$$\leq T(N) := \prod_{j=1}^{m} (c_j \log N) = (\log N)^m \prod_{j=1}^{m} c_j$$

Proof of Sharpness: Counting Representations

- number of different power-products appearing in multi-base representations of $\{1, 2, \dots, N\}$

$$\leq T(N) := \prod_{j=1}^{m} (c_j \log N) = (\log N)^m \prod_{j=1}^{m} c_j$$

- number of representations with weight at most K

$$R_K(N) \leq \sum_{k=1}^{K} \binom{T(N)}{k} (|D| - 1)^k \leq \big(|D|\, T(N)\big)^K$$

Proof of Sharpness: Counting Representations

- number of different power-products
  appearing in multi-base representations of $\{1, 2, \ldots, N\}$

$$\leq T(N) := \prod_{j=1}^{m} (c_j \log N) = (\log N)^m \prod_{j=1}^{m} c_j$$

- number of representations with weight at most K

$$R_K(N) \leq \sum_{k=1}^{K} \binom{T(N)}{k} (|D| - 1)^k \leq \bigl(|D|\,T(N)\bigr)^K$$

- suppose all integers in $\{2^{s-1}+1, 2^{s-1}+2, \ldots, 2^s\}$
  have a representation with weight at most K, i.e.

$$\bigl(|D|\,T(2^s)\bigr)^K \geq R_K(2^s) \geq 2^{s-1}$$

- take logarithms

Different Point of View: Communication Complexity

Communication Complexity

- Set-up:
  - Alice and Bob both hold ℓ bits of information (nonnegative integers less than 2^ℓ)
  - Bob wants to check if both hold the same information

Different Point of View: Communication Complexity

Communication Complexity

- Set-up:
  - Alice and Bob both hold ℓ bits of information (nonnegative integers less than 2^ℓ)
  - Bob wants to check if both hold the same information
  - Alice send some piece of information (according protocol)
  - Bob says
    - "$=$"
    - "$\neq$"
    - "more"

Different Point of View: Communication Complexity

Communication Complexity

- Set-up:
  - Alice and Bob both hold ℓ bits of information (nonnegative integers less than 2^ℓ)
  - Bob wants to check if both hold the same information
  - Alice send some piece of information (according protocol)
  - Bob says
    - "$=$"
    - "$\neq$"
    - "more"
- for each deterministic algorithm/protocol $\rightsquigarrow$ instance where ℓ communication bits needed

Proof of Sharpness: Communication Complexity

- assume n has multi-base representation with only $o\left(\frac{\log n}{\log \log n}\right)$ summands
- convert above $\ell = \lfloor \log n \rfloor$-bit instance to multi-base representation
- summand can be denoted by $\mathcal{O}(\log \log n)$ bits

Proof of Sharpness: Communication Complexity

- assume n has multi-base representation with only $o\left(\frac{\log n}{\log\log n}\right)$ summands
- convert above $\ell = \lfloor \log n \rfloor$-bit instance to multi-base representation
- summand can be denoted by $\mathcal{O}(\log\log n)$ bits
- Alice only needs

$$\mathcal{O}(\log\log n) \cdot o\left(\frac{\log n}{\log\log n}\right) = o(\log n)$$

bits to tell Bob everything

Minimal Hamming Weight

Theorem (K–Suppakitpaisarn–Wagner 2018)

- *fix bases* $p_1, \ldots, p_m$ $(m \geq 2)$ *multiplicatively independent*
- *fix digit set containing* 1
- *there exist positive constants* K_1 *and* K_2
  - (U) *each integers* n *has representation with Hamming weight at most* $K_1 \frac{\log n}{\log \log n}$
  - (L) *infinitely many positive integers* n *with no representation with Hamming weight less than* $K_2 \frac{\log n}{\log \log n}$