Double-Base Chains for Scalar Multiplications on Elliptic Curves

Wei Yu, Saud Al Musa, and Bao Li

Institute of Information Engineering, Chinese Academy of Sciences
yuwei_1_yw@163.com

May, 2020

Abstract

- Introduction

Abstract

- Introduction
- The Number of Double-Base Chains

Abstract

- Introduction
- The Number of Double-Base Chains
- Hamming Weight of Double-Base Chains

Abstract

- Introduction
- The Number of Double-Base Chains
- Hamming Weight of Double-Base Chains
- Dynamic Programming to Generate Optimal Double-Base Chains

Abstract

- Introduction
- The Number of Double-Base Chains
- Hamming Weight of Double-Base Chains
- Dynamic Programming to Generate Optimal Double-Base Chains
- Scalar Multiplication using Double-Base Chains

Introduction:Double-Base Chain

Double-base chains (DBCs) are used to speed up scalar multiplications on elliptic curves.
A DBC represents an integer n as

$$\sum_{i=1}^{l} c_i 2^{b_i} 3^{t_i}$$

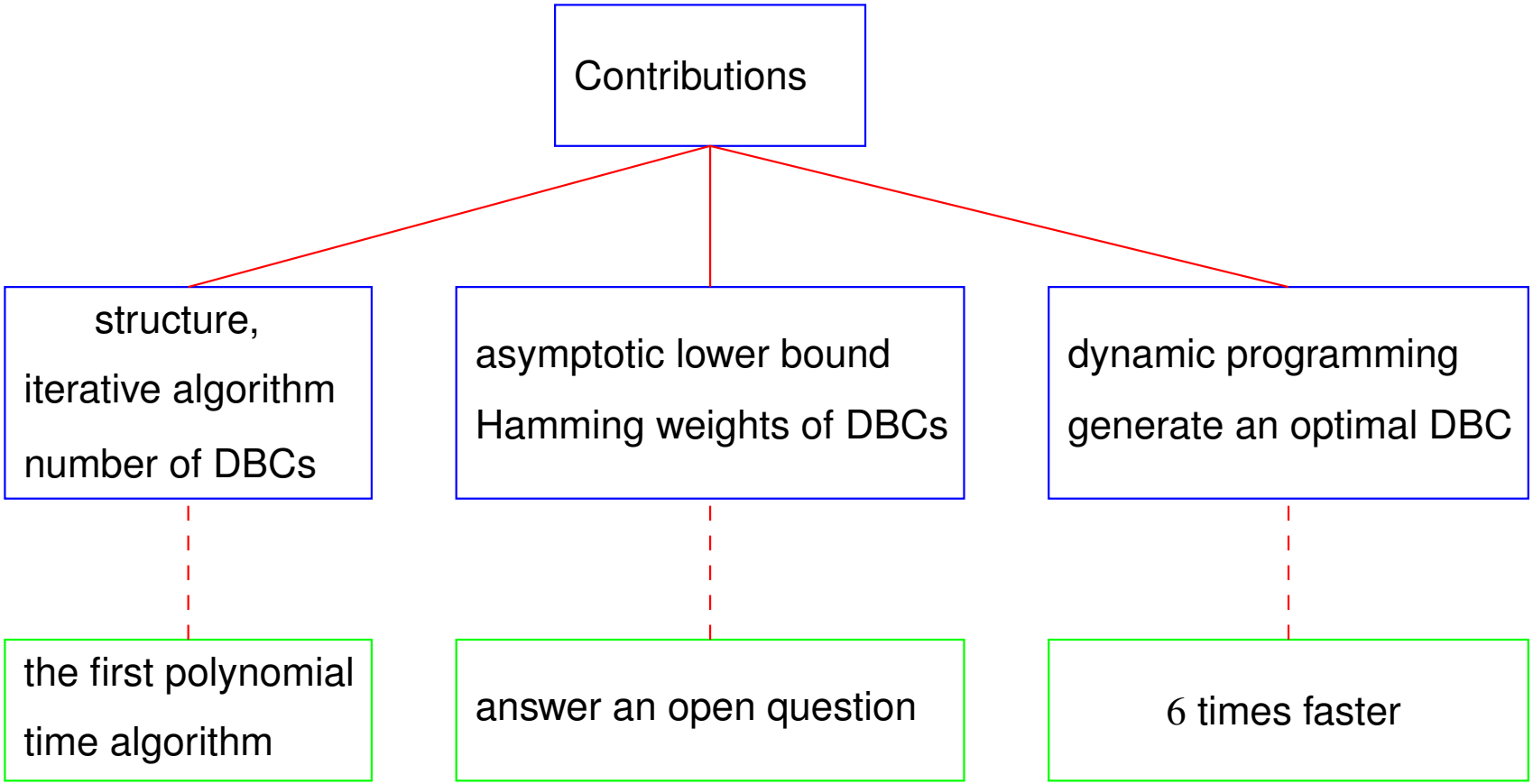
$c_i \in \mathscr{C} = \{\pm 1\}, b_l \geq b_{l-1} \geq \ldots \geq b_1 \geq 0$ and $t_l \geq t_{l-1} \geq \ldots \geq t_1 \geq 0$.

- $2^{b_i} 3^{t_i}$: a term
- $2^{b_l} 3^{t_l}$: the leading term
- l: the Hamming weight

Introduction:Double-Base Chain

1. Dimitrov, Imbert, and Mishra: The canonic DBCs of a positive integer n are the ones with minimal Hamming weight.
2. An optimal DBC of n is the DBC with the smallest value in the set $\{\mathrm{val}(\mathrm{w})|\mathrm{w} \in X\}$ where X is the set containing all DBCs of n. w is defined by the cost of scalar multiplication.

Introduction:Contributions



The Number of DBCs

Counting the number of DBCs:

1. To show DBC is redundant
2. To generate an optimal DBC

Each positive integer has at least one DBC such as binary representation.
Imbert and Philippe 2010: an elegant algorithm to compute the number of unsigned DBCs for a given integer.

The Number of DBCs

Doche 2014

1. calculate the number of DBCs with a leading term dividing 2^b3^t for a positive integer
2. efficient for less than $70-$bit integers with a leading term dividing 2^b3^t for the most b and t
3. exponential time algorithm

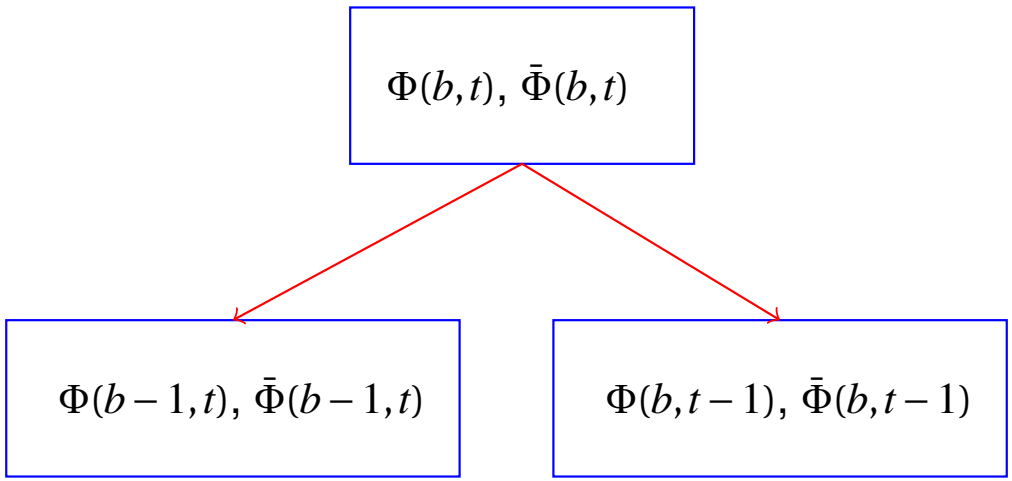
The Number of DBCs:The Structure of the Set Containing All DBCs

$\Phi(b,t,n)$: the set containing all DBCs of an integer $n \geq 0$ with a leading term strictly dividing $2^b 3^t$.
$\bar{\Phi}(b,t,n)$: the set containing all DBCs of an integer $n \leq 0$ with a leading term strictly dividing $2^b 3^t$.

The Number of DBCs:The Structure of the Set Containing All DBCs

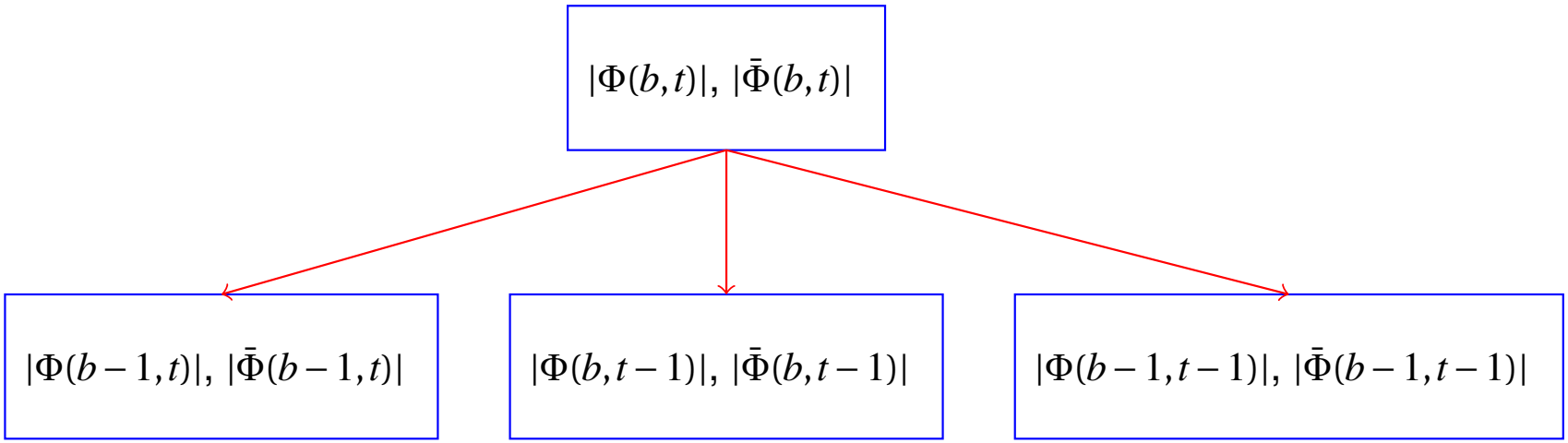
Let n be a positive integer, $b \geq 0$, $t \geq 0$, and $b + t > 0$. The structure of $\Phi(b,t)$ and that of $\bar{\Phi}(b,t)$ are described as follows.

Figure: The Structure of DBCs



The Number of DBCs:The Structure of the Set Containing All DBCs

Figure: The Cardinality of the Set Containing All DBCs



The Number of DBCs: Iterative Algorithm

Input: A positive integer n, $b \geq 0$, and $t \geq 0$
Output: $|\Phi(b,t)|$, $|\bar{\Phi}(b,t)|$

1. $|\Phi(0,0)| \leftarrow 1$, $|\bar{\Phi}(0,0)| \leftarrow 0$
2. **For** i **from** 0 **to** b, $|\Phi(i,-1)| = |\bar{\Phi}(i,-1)| \leftarrow 0$
3. **For** j **from** 0 **to** t, $|\Phi(-1,j)| = |\bar{\Phi}(-1,j)| \leftarrow 0$
4. **For** j **from** 0 **to** t
5. **For** i **from** 0 **to** b
6. **If** $i+j>0$, compute $|\Phi(i,j)|$ and $|\bar{\Phi}(i,j)|$
7. **return** $|\Phi(b,t)|$, $|\bar{\Phi}(b,t)|$

The time complexity of our iterative algorithm is in $\mathcal{O}\left((\log n)^3\right)$ bit operations when both b and t are $\mathcal{O}(\log n)$.

The Number of DBCs

1. 100 has 2590 DBCs with a leading term dividing $2^{30}3^4$.
2. 1000 has 28364 DBCs with a leading term dividing $2^{30}3^6$.
3. the number of DBCs of $\lfloor \pi \times 10^{120} \rfloor$ with a leading term dividing $2^{240}3^{120}$ is
$$405694512689803328570475272448020332384436179545046727328115784367271984621308621154227072670259226179970361\ 05303878574879.$$

Hamming Weight of DBCs:Open Question

Open question

Whether the average Hamming weight of DBCs produced by the greedy approach is $\mathcal{O}\left(\frac{\log n}{\log\log n}\right)$ or not

Doche and Habsieger 2008

Hamming Weight of DBCs

Efforts to investigate the lower bound of DBCs

1. Dimitrov and Howe: there exist infinitely many integers n whose shortest double-base number system representations have Hamming weights $\Omega\left(\frac{\log n}{\log\log n \log\log\log n}\right)$.
2. Lou, Sun, and Tartary: there exists at least one $\lfloor \log n \rfloor$ –bit integer such that any DBC representing this integer needs at least $\Omega(\lfloor \log n \rfloor)$ terms.

Hamming Weight of DBCs

1. The number of DBCs of a positive integer is infinite
2. The leading term of its DBC may be infinite

Hamming Weight of DBCs:The Range of the Leading Term of Optimal DBCs and Canonic DBCs

Disanto, Imbert, and Philippe 2014 showed $2^{b_l}3^{t_l} > \frac{n}{2}$.
Let n be a positive integer represented as a DBC.
This work shows

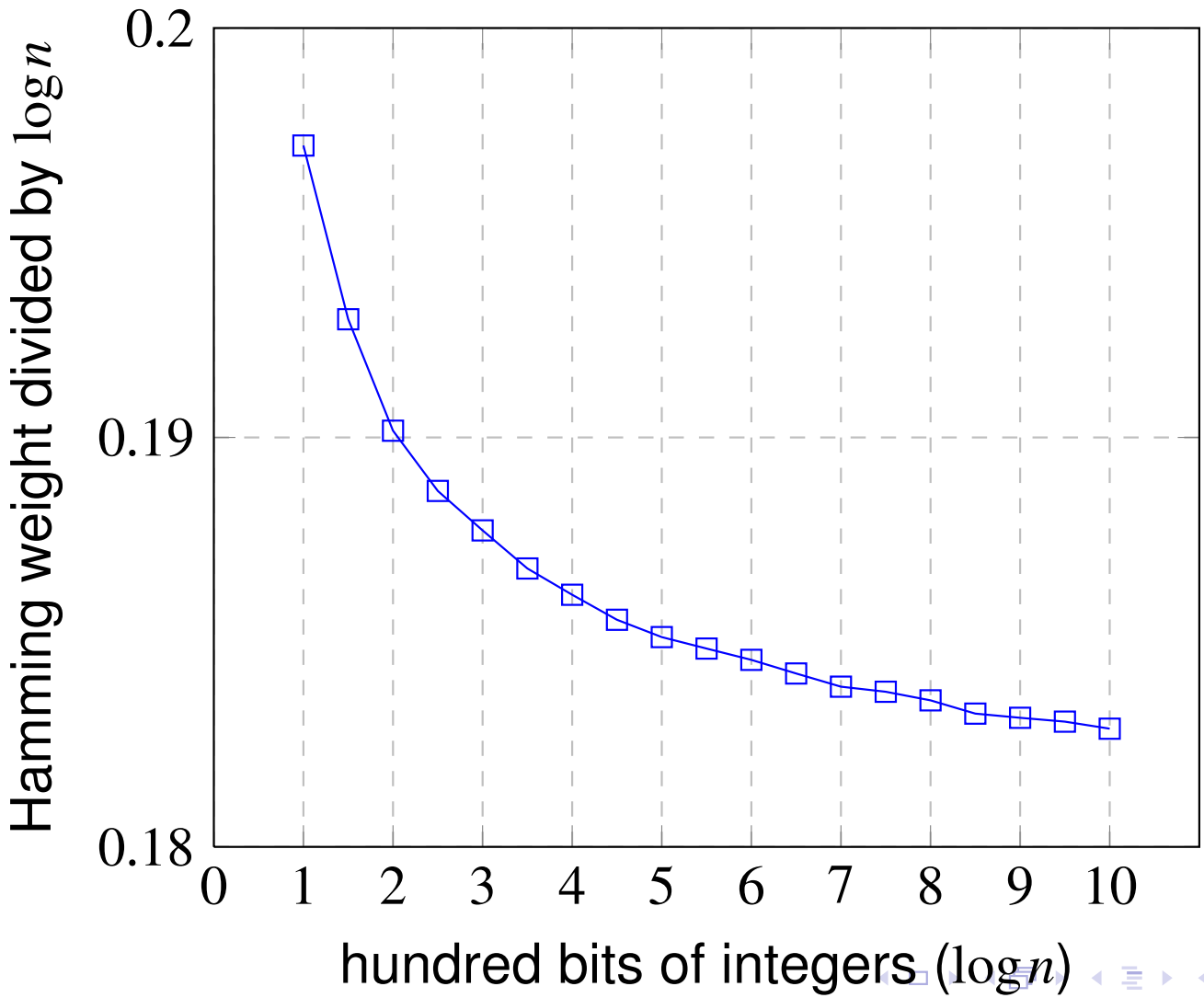
1. $\frac{n}{2} < 2^{b_l}3^{t_l} < 2n$ when w is an optimal DBC
2. $\frac{16n}{21} < 2^{b_l}3^{t_l} < \frac{9n}{7}$ when w is a canonic DBC

Hamming Weight of DBCs

An asymptotic lower bound of the average Hamming weights of canonic DBCs for $(\log n)$–bit integers is $\frac{\log n}{8.25}$.
This answers Doche's open question.

Hamming Weight of DBCs

Figure: The Hamming weight of canonic DBCs of integers



Hamming Weight of DBCs

1. $0.182887 \log n$ for $1000-$bit integers,
2. $0.181101 \log n$ for $2000-$bit integers,
3. $0.179822 \log n$ for $3000-$bit integers.

This value of the Hamming weight given for $3000-$bit integers still has a distance from the theoretical lower bound $\frac{\log n}{8.25}$.

Dynamic Programming Algorithm to Produce Optimal DBCs

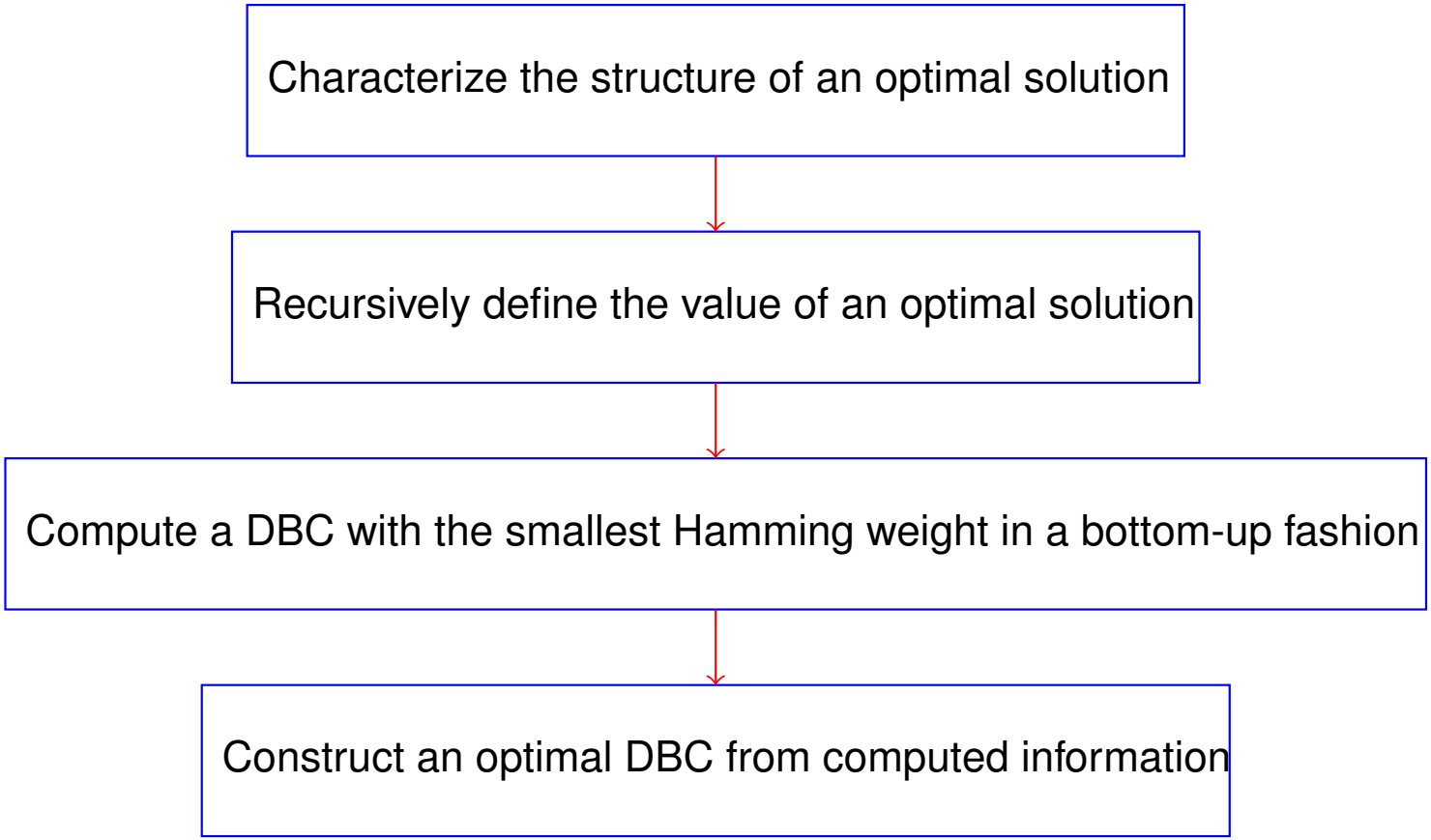
algorithm	time complexity ($\mathcal{O}$)	space complexity ($\mathcal{O}$)
Doche 2014	exponential	$(\log n)^2$
Capuñay and Thériault 2015	$(\log n)^4$	$(\log n)^3$
Bernstein, Chuengsatiansup, and Lange 2017	$(\log n)^{2.5}$	$(\log n)^{2.5}$
Dynamic Programming (new)	$(\log n)^2 \log\log n$	$(\log n)^2$

Dynamic Programming Algorithm to Produce Optimal DBCs

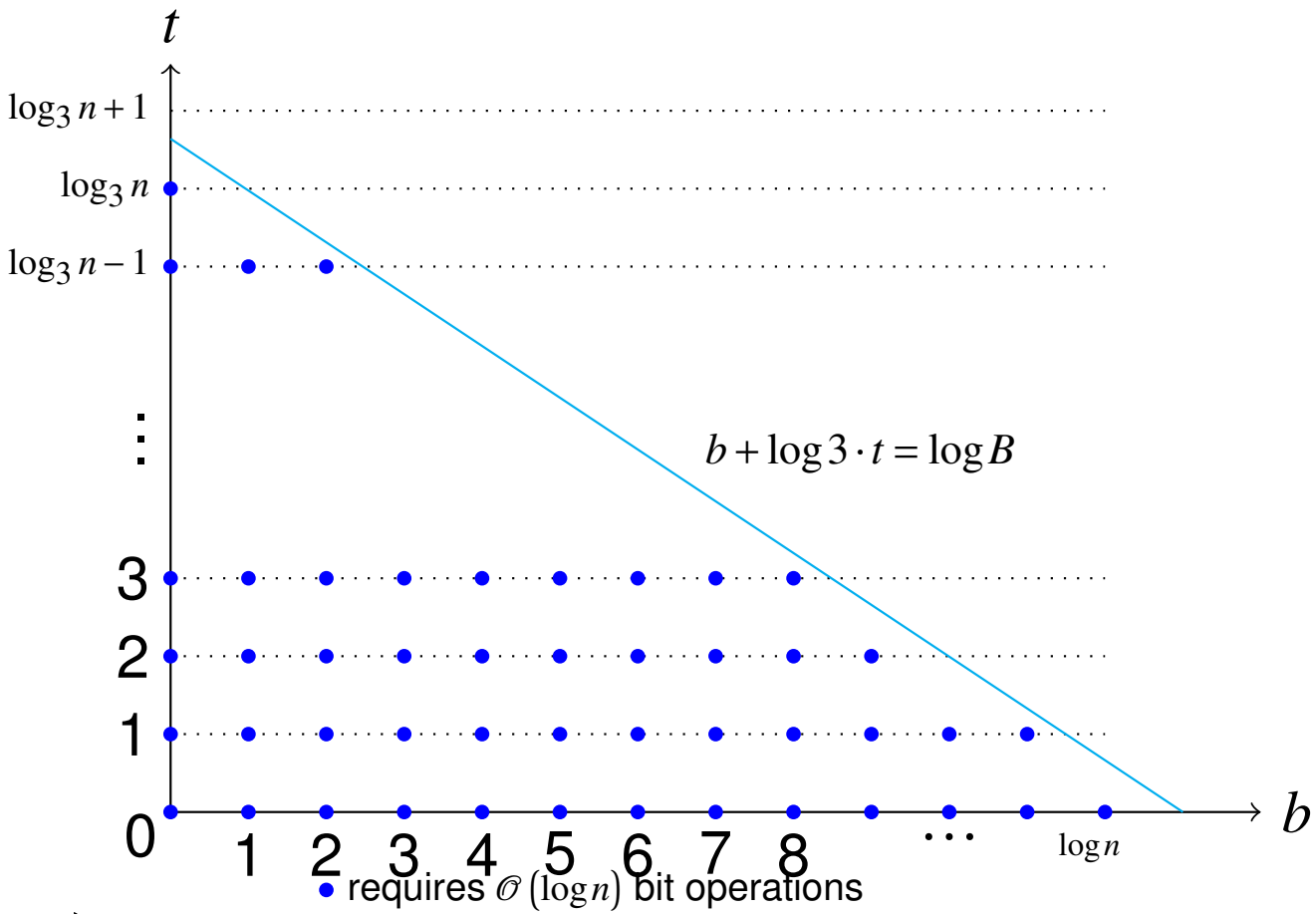
Dynamic programming solves problems by combining the solutions of subproblems.
Two key characteristics

1. optimal substructure
2. overlapping subproblems

Dynamic Programming Algorithm to Produce Optimal DBCs:Blueprint

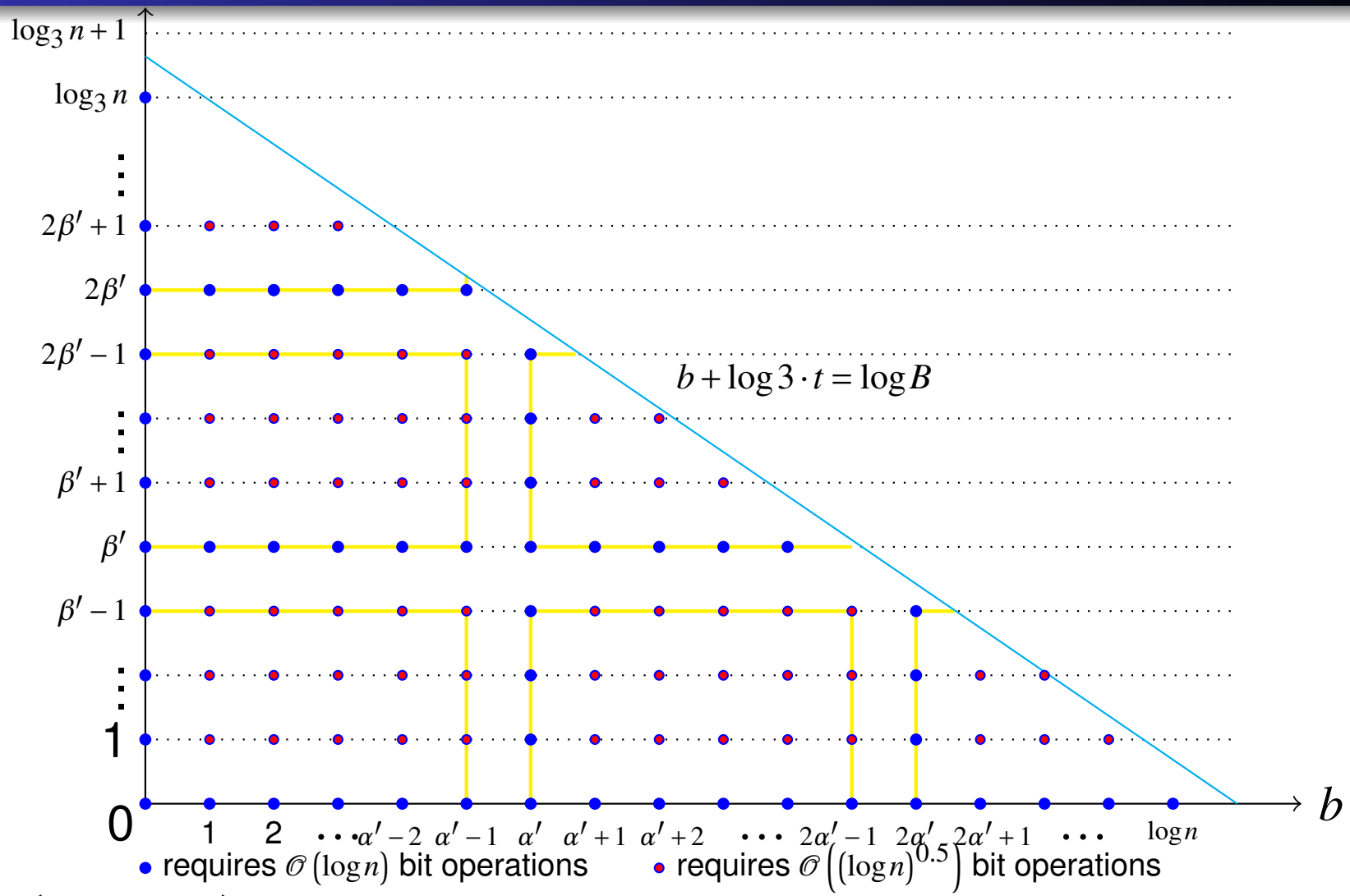


Dynamic Programming Algorithm to Produce Optimal DBCs



$\mathcal{O}\left((\log n)^3\right)$ bit operations

Dynamic Programming Algorithm to Produce Optimal DBCs:Reduced Representatives

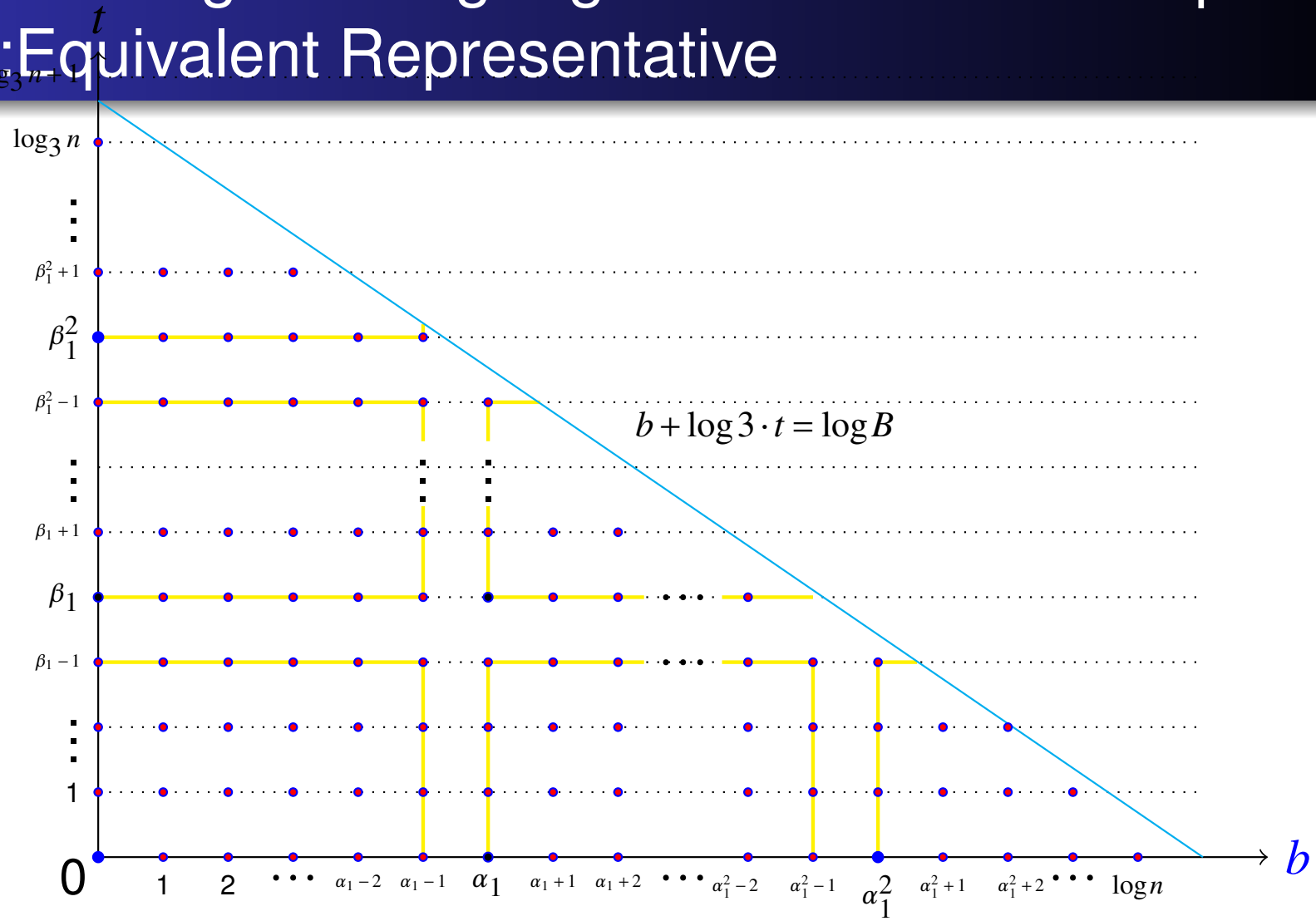


$\mathcal{O}\left((\log n)^{2.5}\right)$ bit operations

Dynamic Programming Algorithm to Produce Optimal DBCs:Equivalent Representative

Bernstein, Chuengsatiansup, and Lange's reduced representatives for large numbers do not work for $\log n + \log_3 n$ boundary nodes.
Our equivalent representatives will solve this problem.

Dynamic Programming Algorithm to Produce Optimal DBCs:Equivalent Representative



- requires $\mathcal{O}(\log n)$ bit operations
- requires $\mathcal{O}\left((\log n)^{2/3}\right)$ bit operations
- requires $\mathcal{O}\left((\log n)^{1/3}\right)$ bit operations

$\mathcal{O}\left((\log n)^{7/3}\right)$ bit operations

Dynamic Programming Algorithm to Produce Optimal DBCs:Equivalent Representative Repeatedly

Our dynamic programming algorithm terminates in $\mathcal{O}(\log^2 n \log\log n)$ bit operations and in $\mathcal{O}(\log^2 n)$ bits of memory.

Dynamic Programming Algorithm to Produce Optimal DBCs: Comparisons

Our dynamic programming algorithm

1. over 20 times faster than Capuñay and Thériault's algorithm
2. over 6 times faster than Bernstein, Chuengsatiansup and Lange's algorithm

As the integer becomes larger, our dynamic programming algorithm will gain more.

Scalar Multiplication using Double-Base Chains

1. Edwards curve
2. Weierstrass curve
3. Tripling-oriented Doche-Icart-Kohel curves (DIK curves)

Bernstein, Lange: Explicit-formulas database.

Table: Cost of elliptic curve point operations

curve	mA	D	T
Weierstrass	7**M**+4**S**	3**M**+5**S**	7**M**+7**S**
Edwards	8**M**+4**S**	3**M**+4**S**	9**M**+4**S**
DIK	7**M**+4**S**	2**M**+7**S**	6**M**+6**S**

Scalar Multiplication using Double-Base Chains:Improvement

Table: Cost of elliptic curve point operations

curve	improvement to NAF
Edwards	10%
Weierstrass	13%
DIK	20%

Conclusion

1. The theoretical aspects of DBCs arising from their study to speed up scalar multiplication
2. Producing an optimal DBC efficiently.

Any questions please send email to:yuwei_1_yw@163.com
Thanks for your time!