KDDI Research

符号暗号の高速求解手法の実装に向けて

Towards the fast decoding algorithms for code-based cryptosystems

2022年8月3日
KDDI総合研究所
成定 真太郎

目次

背景

- アメリカ標準技術研究所（NIST）の耐量子暗号に関する標準化プロジェクト：
  NIST-PQC (post-quantum cryptography)
- 耐量子暗号の鍵共有/署名アルゴリズムとして、4手法が選定・4手法がRound 4に進出

: Selected　　: Round 4

格子暗号	符号暗号	同種写像暗号	ハッシュ
CRYSTALS-KYBER (Selected)	Classic McEliece (Round 4)	SIKE (Round 4)	SPHINCS+ (Selected)
CRYSTALS-DILITHIUM (Selected)	BIKE (Round 4)		
FALCON (Selected)	HQC (Round 4)		

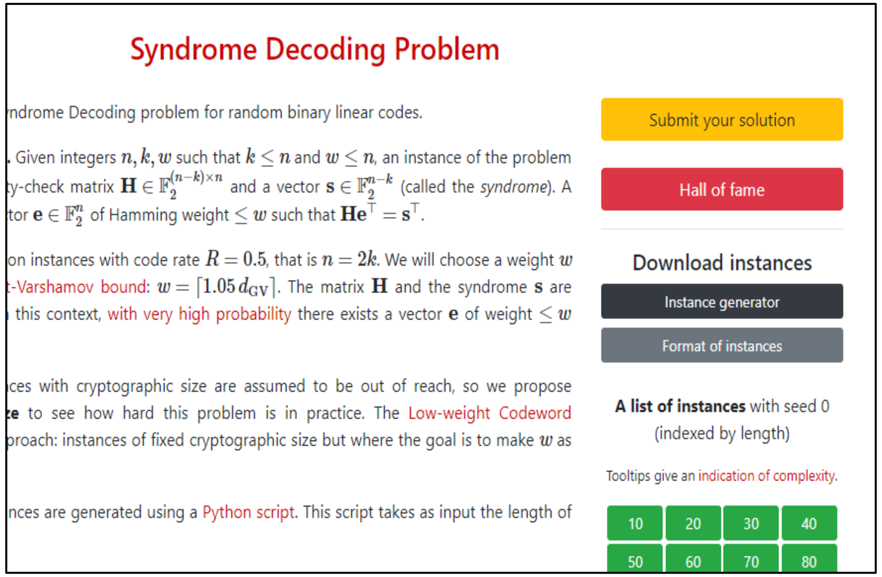
- 標準化の可能性がある符号暗号方式に着目

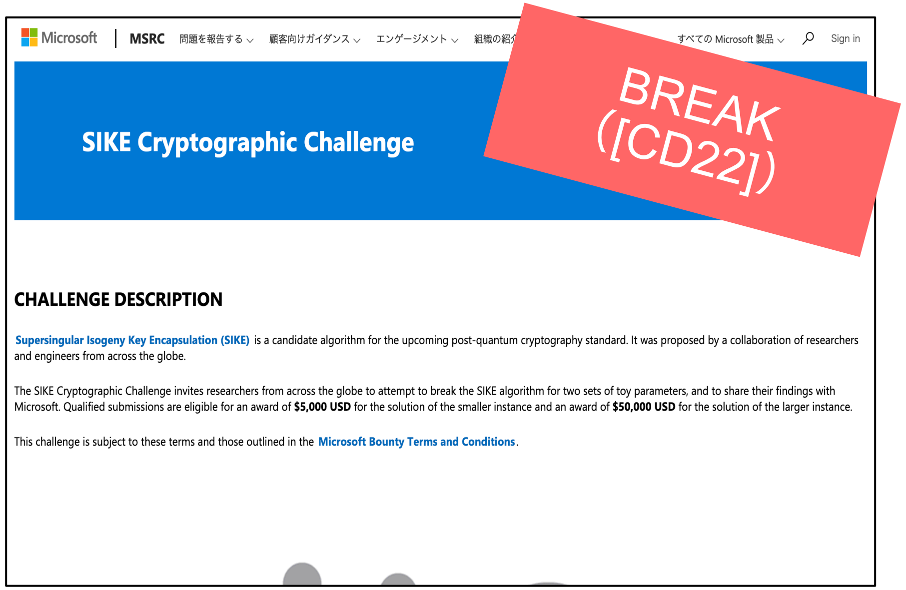
> "Although Classic McEliece is widely regarded as secure, NIST does not anticipate it being widely used due to its large public key size. **NIST may choose to standardize Classic McEliece** at the end of the fourth round." *NIST PQC Forum 2022/7/6

背景

- 各国の研究機関が主催する**暗号解読コンテスト**を通して実用面での安全性を評価
  - ➡ より高次元の暗号を解読することで暗号の最適(**安全・高速**)なパラメータを設計可能

SVP Challenge（格子）2013 ~	Decoding Challenge（符号） 2019 ~	SIKE Cryptographic Challenge（同種写像） 2021 ~





符号暗号とシンドローム復号問題 (SDP)

KDDI Research

Syndrome Decoding Problem (SDP)

入力：整数n, k, w、行列$\boldsymbol{H} \in \mathbb{F}_2^{(n-k)\times n}$およびベクトル $\boldsymbol{s} \in \mathbb{F}_2^{n-k}$

出力：$\boldsymbol{He} = \boldsymbol{s}$を満たすベクトル $\boldsymbol{e} \in \mathbb{F}_2^{n}$　ただし、$\mathrm{wt}(\boldsymbol{e}) = w$

$SDP(n, k, w)$と書く

$n = 8$
$k = 4$
$\mathrm{w} = 3$

n本のベクトル：公開鍵

$\boldsymbol{s}$：公開鍵

$$
\begin{array}{cccccccc}
\boldsymbol{h}_1 & \boldsymbol{h}_2 & \boldsymbol{h}_3 & \boldsymbol{h}_4 & \boldsymbol{h}_5 & \boldsymbol{h}_6 & \boldsymbol{h}_7 & \boldsymbol{h}_8
\end{array}
\qquad \boldsymbol{s}
$$

$$
\begin{bmatrix}
1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0
\end{bmatrix}
\qquad
\begin{bmatrix}
1 \\ 1 \\ 1 \\ 1
\end{bmatrix}
$$

選択するベクトルの本数
$w = 3$

$\boldsymbol{e} = 01110000$
(秘密鍵)

$\oplus$

➢ 指数的に多数の解の候補：

$$\binom{n}{w} \approx 2^{nH(w/n)}$$

目次

KDDI Research

Information Set Decoding (ISD)

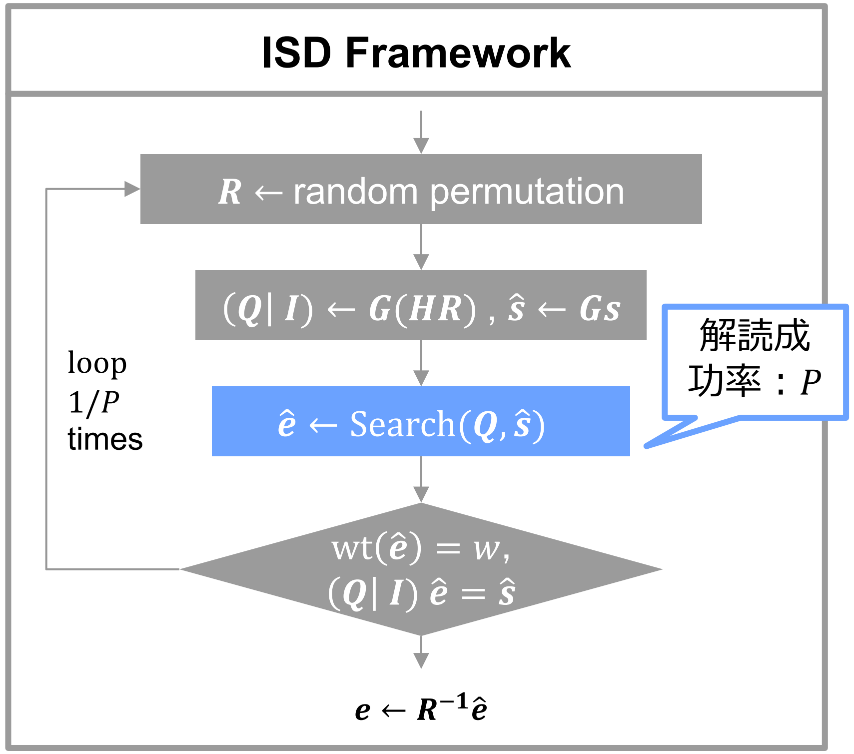
KDDI Research

- ISD: 線形代数・組み合わせ論に基づくSDPの求解アルゴリズムの総称
- 本発表ではISDの中でもメジャーな以下の6手法を紹介

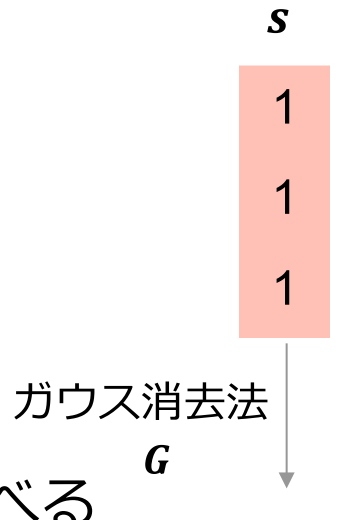
アルゴリズム（著者名）	略称	年度	漸近計算量
Prange [Pra62]	–	1962	$2^{0.121n}$
Dumer [Dum91]	–	1991	$2^{0.117n}$
May-Meurer-Thomae [MMT11]	MMT	2011	$2^{0.112n}$
Becker-Joux-May-Meurer [BJMM12]	BJMM	2012	$2^{0.102n}$
May-Ozerov [MO15]	MO	2015	$2^{0.0953n}$
Both-May [BM18]	BM	2018	$2^{0.0885n}$

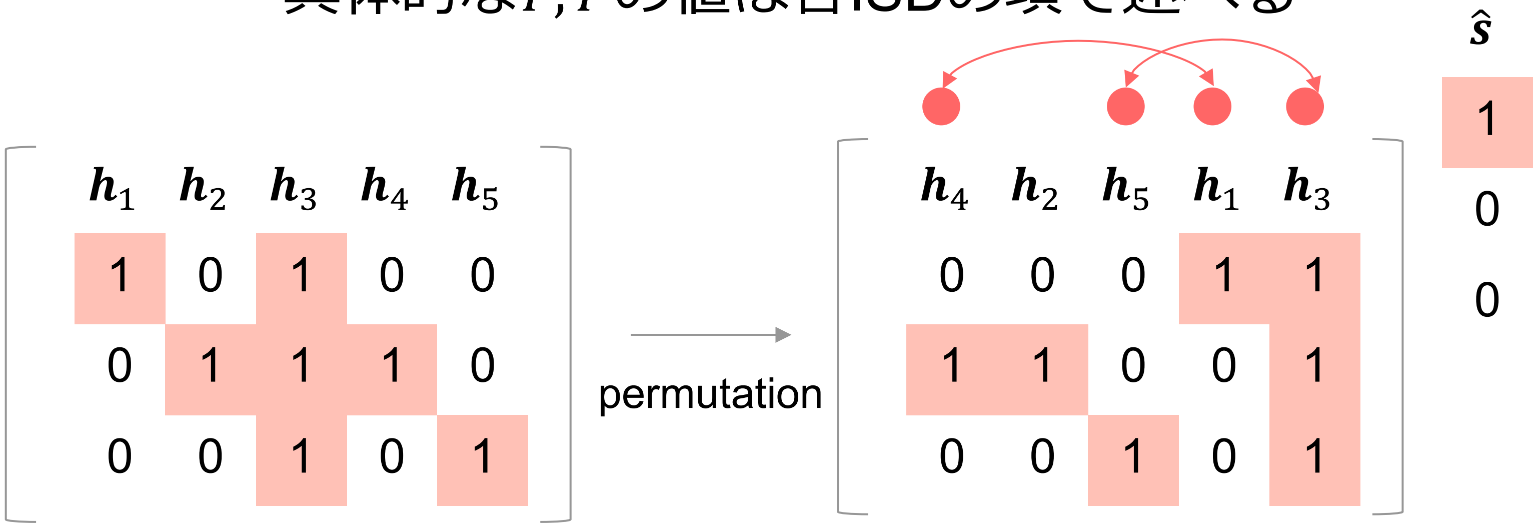
ISDの共通フレームワーク

■ $\boldsymbol{H}$と$\boldsymbol{s}$に対して、ISDはパーミュテーション➔ガウスの消去法➔解の探索を繰り返し行う



■ ISDの計算量(Work Factor)：
- 一度の探索における解読成功率：P
- 一回のloopの計算量：T
- $\mathrm{WF} = T/P$
- 具体的なT, Pの値は各ISDの項で述べる

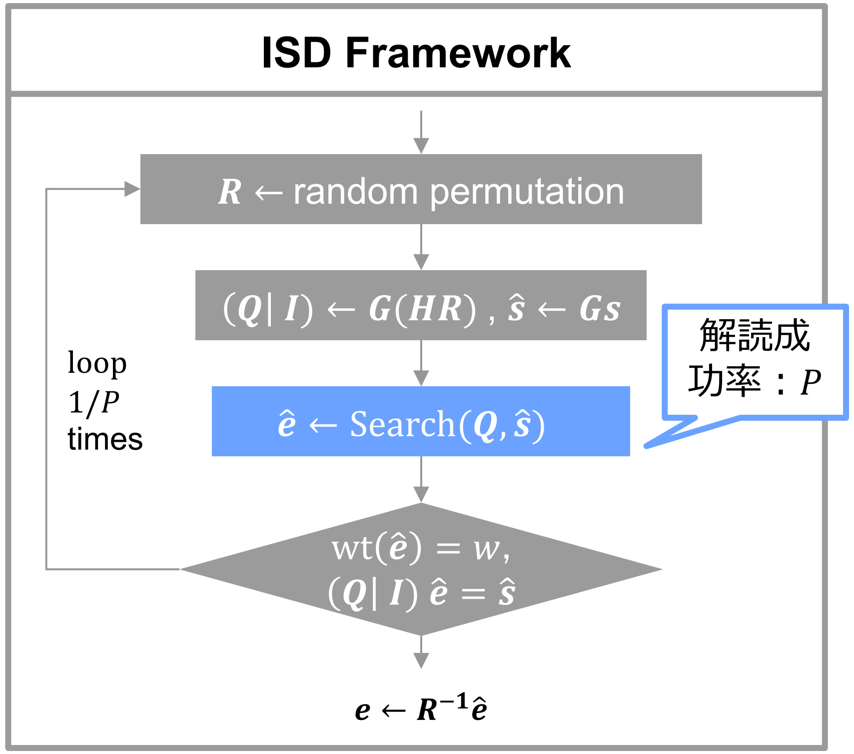


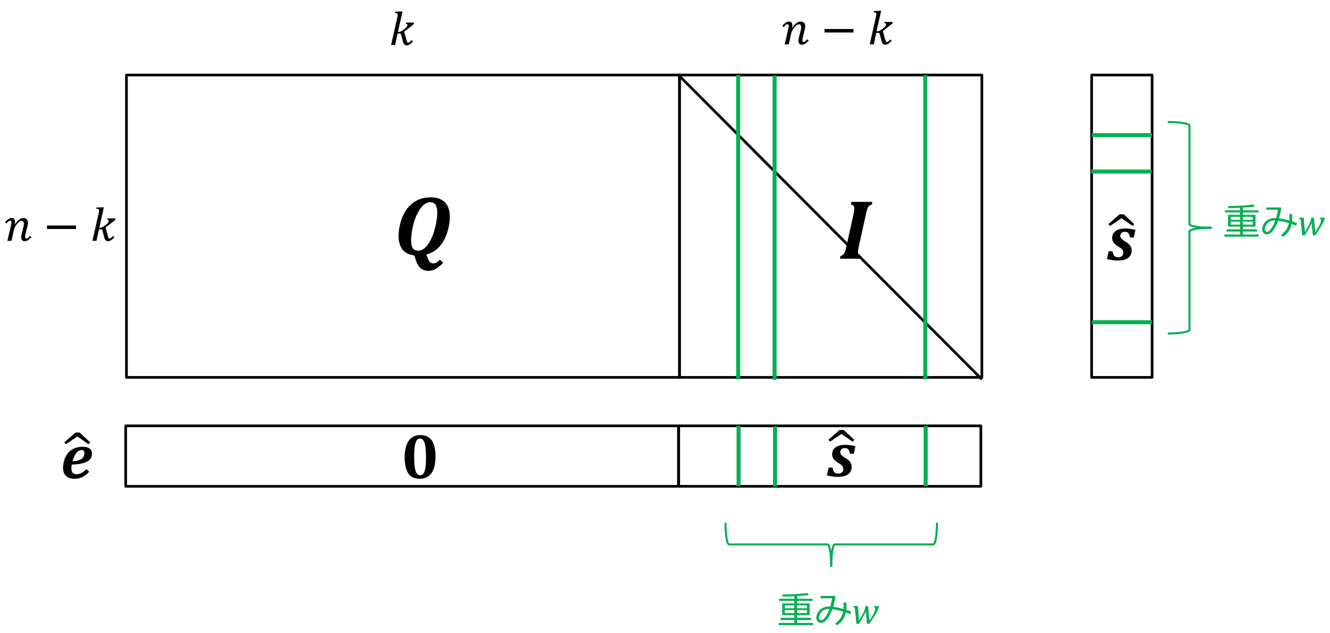


目次

Prange

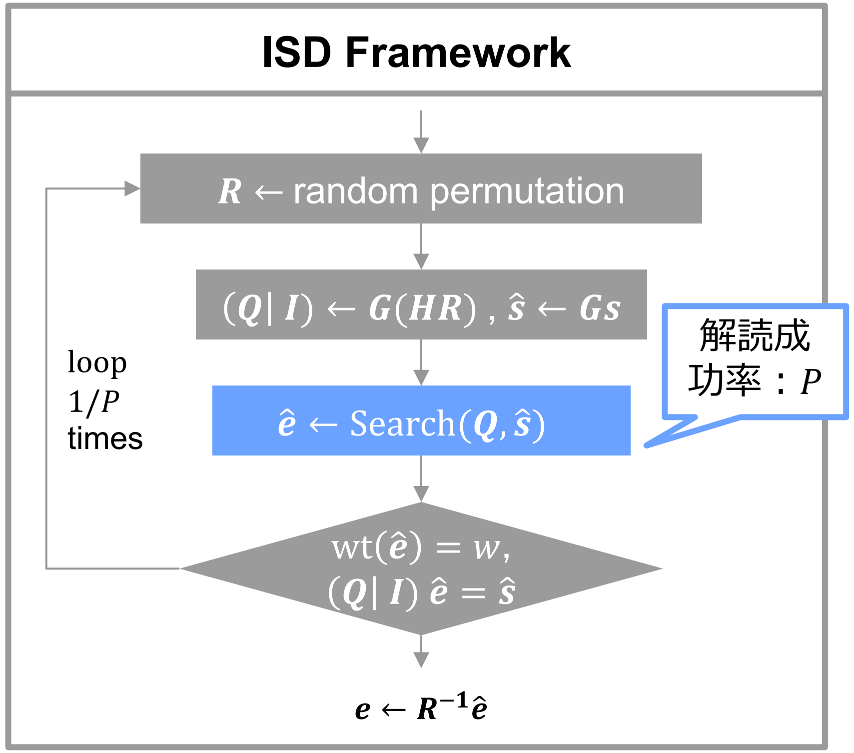
- Prangeは探索は行わず、$\hat{\boldsymbol{s}}$の重みを確認する
- $\mathrm{wt}(\hat{\boldsymbol{s}}) = w$なら$\hat{\boldsymbol{e}} \leftarrow (\boldsymbol{0}, \hat{\boldsymbol{s}})$、$\boldsymbol{e} \leftarrow \boldsymbol{R}^{-1}\hat{\boldsymbol{e}}$が解



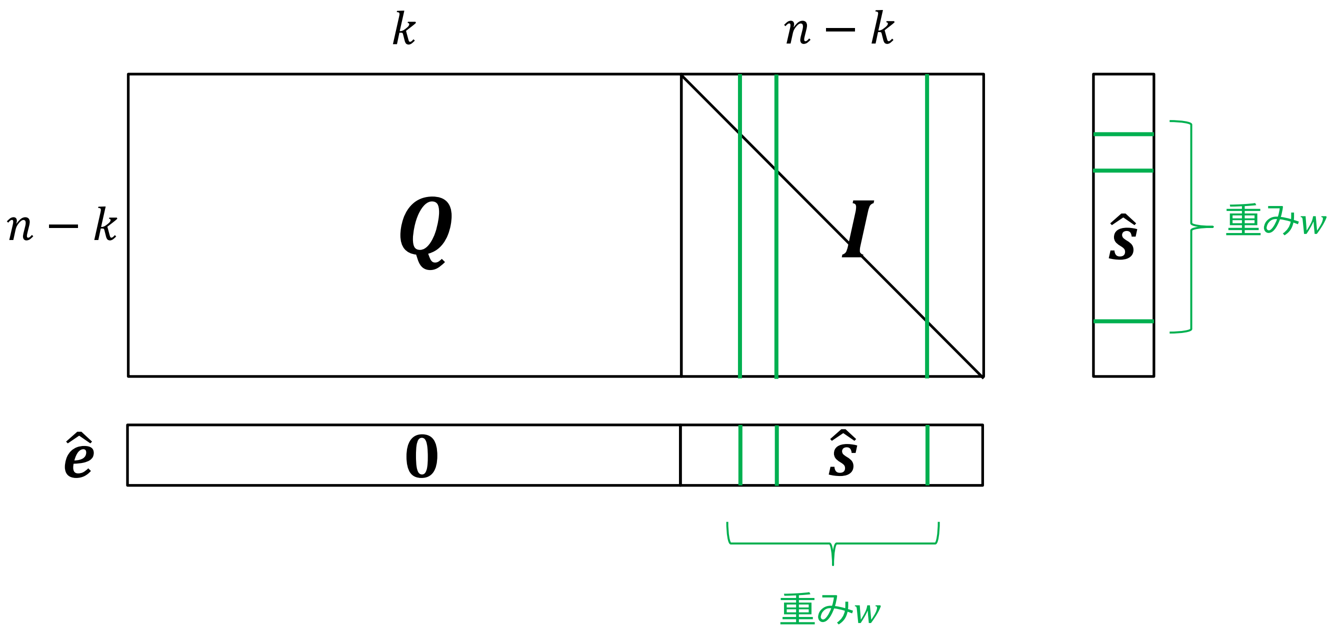


PrangeのWF

- 一度のloopの計算量$T = n(n-k)$：パーミュテーションとガウス消去法の計算量
- 一度の探索での解読成功率$P = \frac{\binom{n-k}{w}}{\binom{n}{w}}$

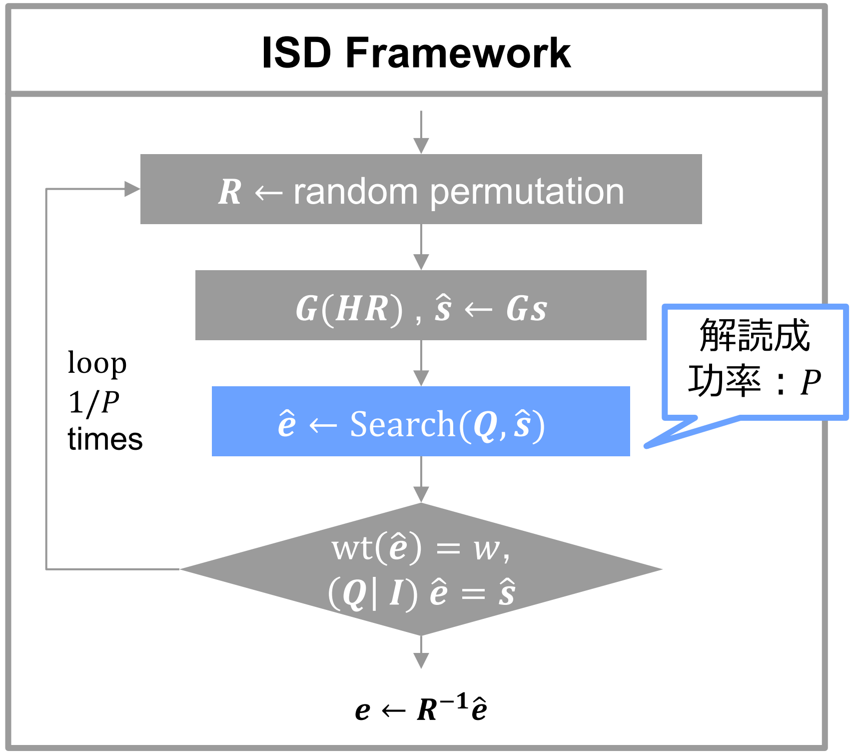


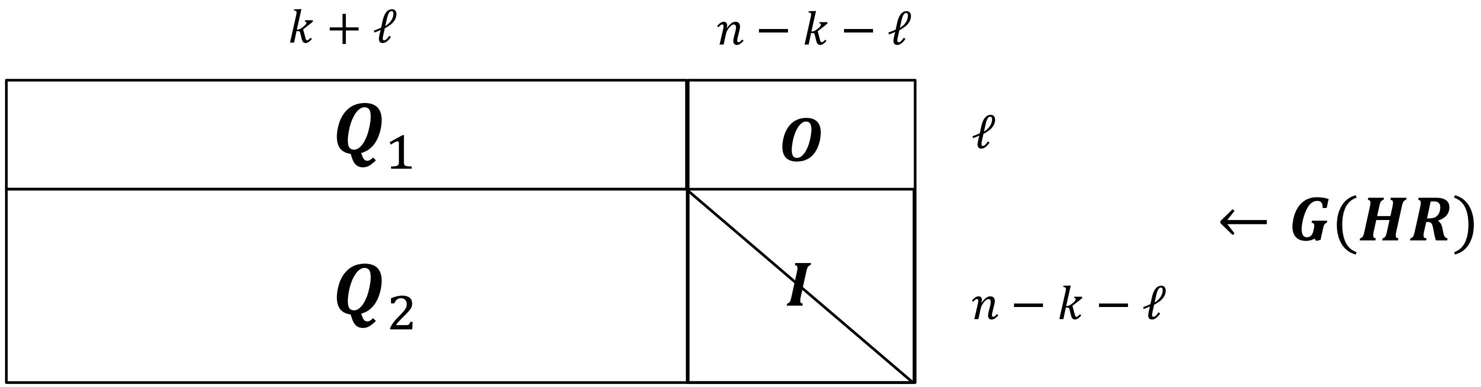
- PrangeのWF $= n(n-k)\frac{\binom{n}{w}}{\binom{n-k}{w}}$



目次

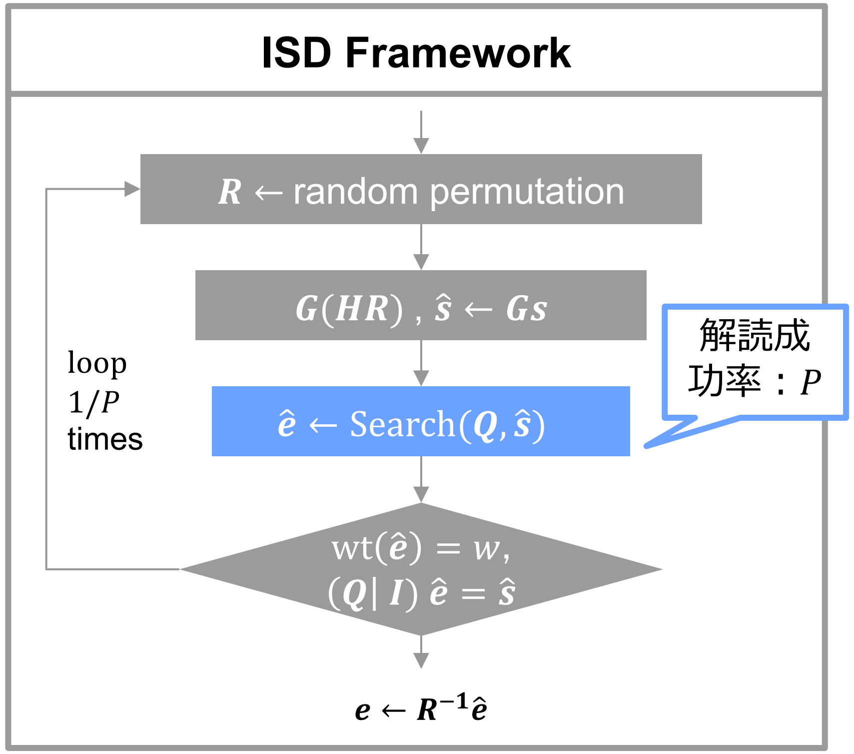
- リスト突合(BirthdayDecode)を用いてWFを削減
- リスト突合のため、$\boldsymbol{H}$を次のように変形





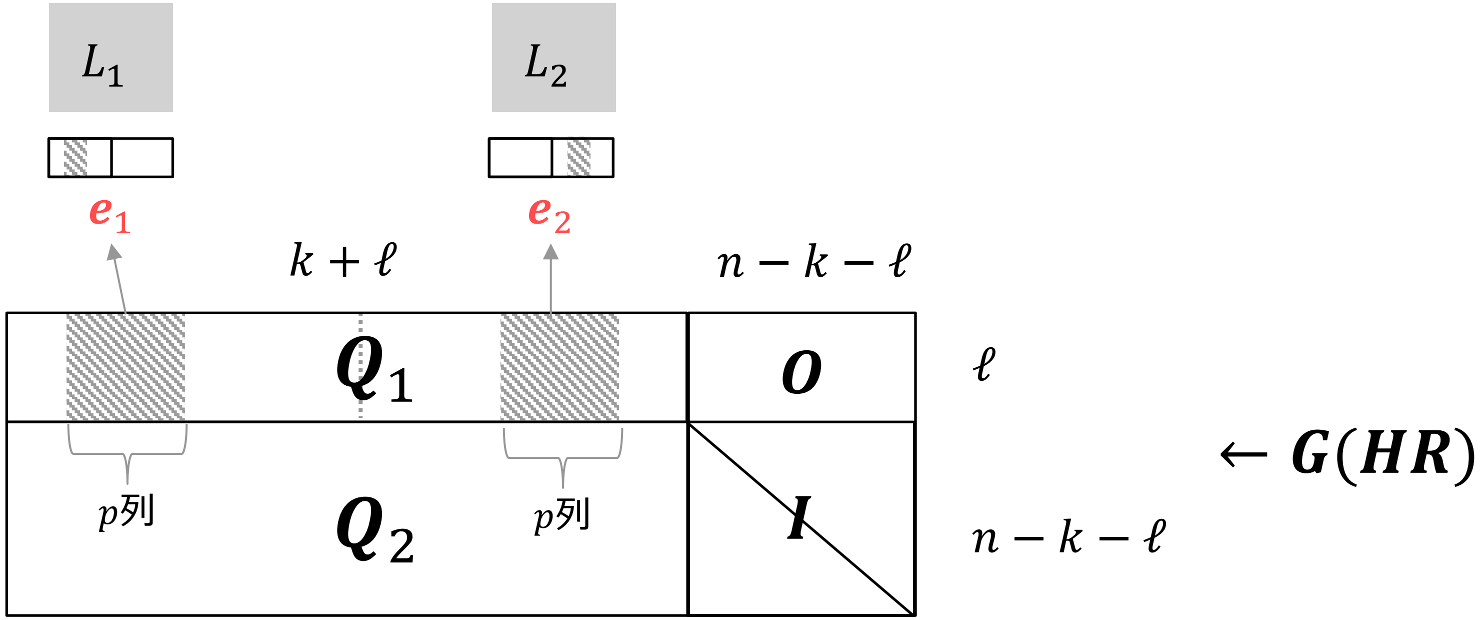
Dumer

- $\boldsymbol{Q}_1$の左右半分から$p \leq w/2$列の数え上げを行い、2つのリストL_1, L_2を構築
- リストはp列の位置$\boldsymbol{e}_1$ ($\boldsymbol{e}_2$)とXORの結果$\mathbf{Q}_1\boldsymbol{e}_1$ ($\mathbf{Q}_1\boldsymbol{e}_2 + \hat{\boldsymbol{s}}_{[\ell]}$)を保持



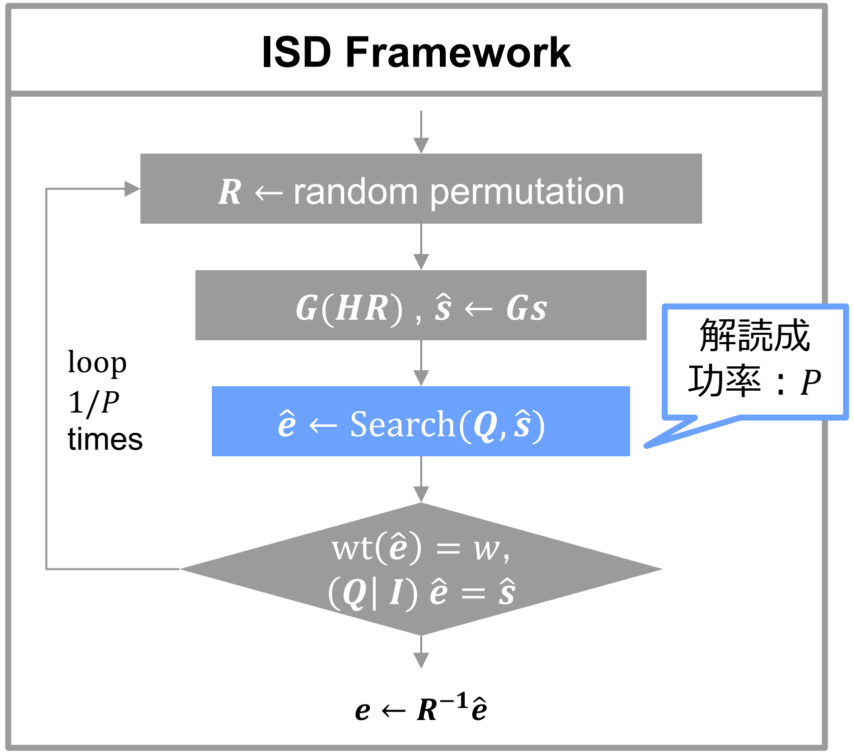
$$
\begin{aligned}
L_1 &= \{(\boldsymbol{e}_1, \boldsymbol{Q}_1\boldsymbol{e}_1) \mid \boldsymbol{e}_1 = (\boldsymbol{a}, 0), \boldsymbol{a} \in I\},\\
L_2 &= \left\{\left(\boldsymbol{e}_2, \boldsymbol{Q}_1\boldsymbol{e}_2 + \hat{\boldsymbol{s}}_{[\ell]}\right) \middle| \boldsymbol{e}_2 = (0, \boldsymbol{a}), \boldsymbol{a} \in I\right\},
\end{aligned}
$$

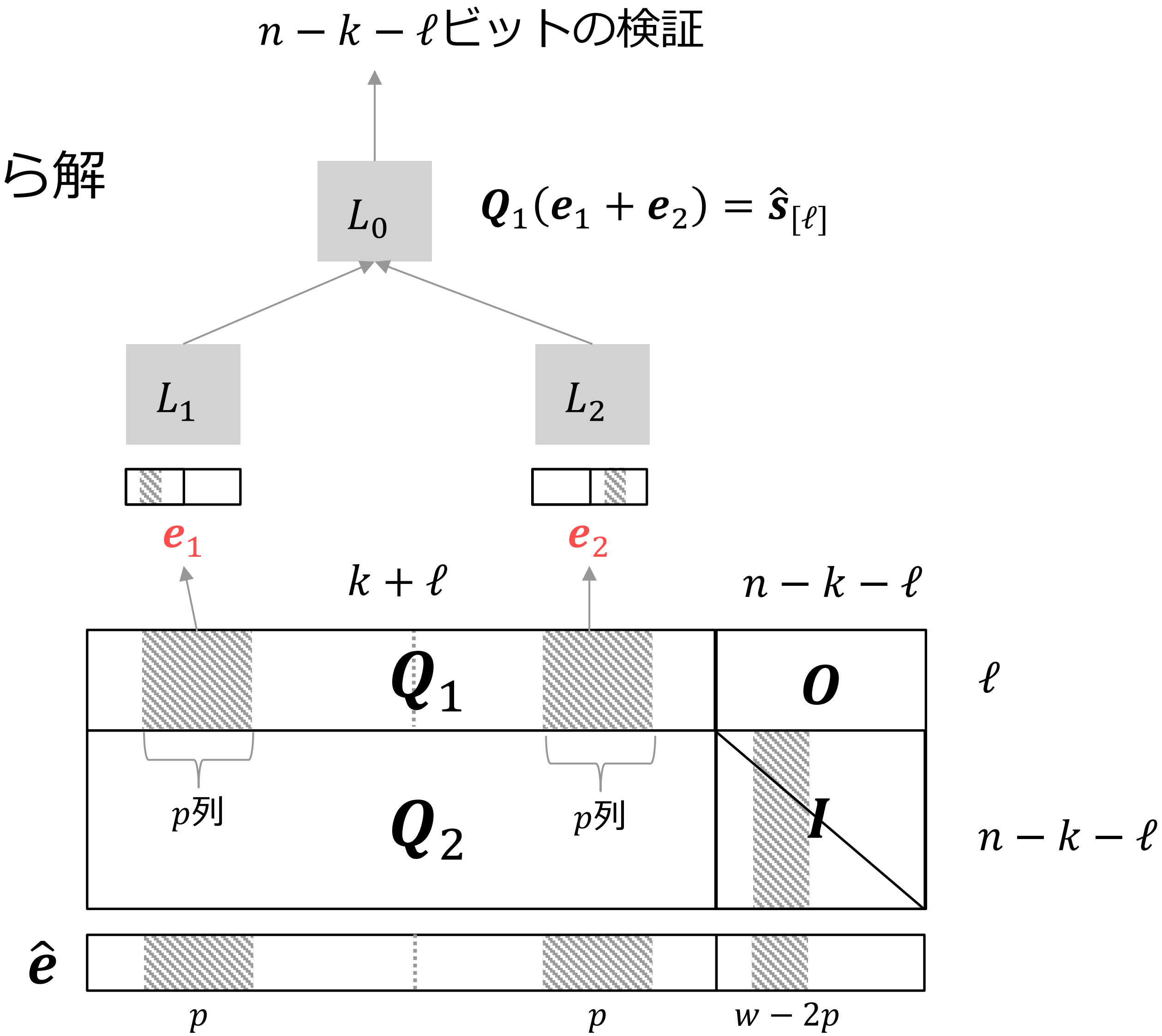
$$
I = \left\{\boldsymbol{a} \in \mathbb{F}_2^{\frac{k+\ell}{2}} \middle| wt(\boldsymbol{a}) = p\right\}
$$



Dumer

- リストL_1, L_2を突合(BirthdayDecode)
  - $\boldsymbol{Q}_1\boldsymbol{e}_1 = \boldsymbol{Q}_1\boldsymbol{e}_2 + \hat{\boldsymbol{s}}_{[\ell]}$
- $\mathrm{wt}(\boldsymbol{Q}_2(\boldsymbol{e}_1 + \boldsymbol{e}_2) + \hat{\boldsymbol{s}}_{[\ell+1,n-k]}) = w - 2p$なら解





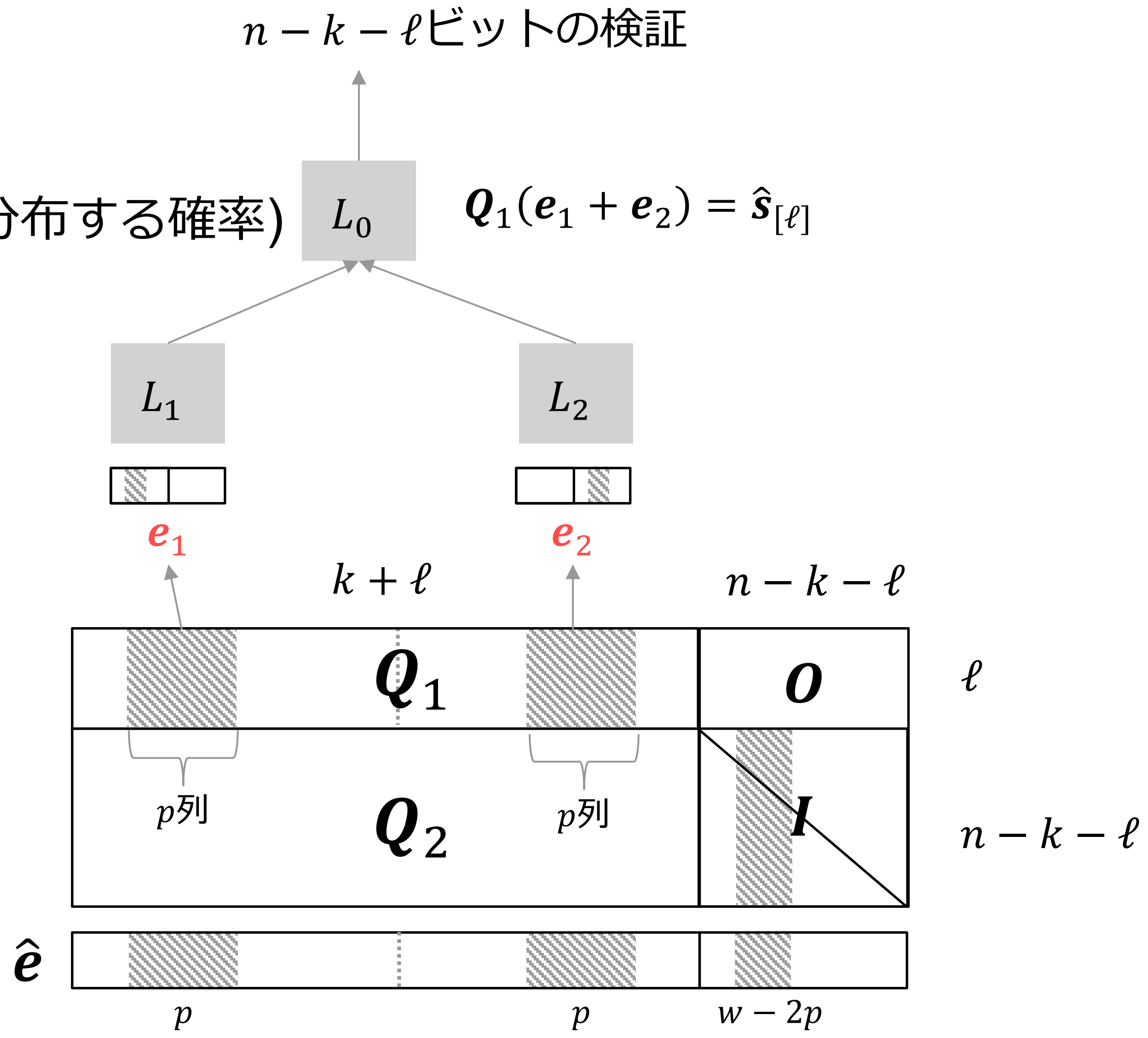
DumerのWF

- $P = \dfrac{\binom{(k+\ell)/2}{p}^2 \binom{n-k-\ell}{w-2p}}{\binom{n}{w}}$

  ($\hat{\boldsymbol{e}}$中のw個の"1"が$\hat{\boldsymbol{e}}$の区間に$p, p, w-2p$で分布する確率)

- $T = n(n-k) + \underbrace{|L_1|}_{|L_1| = \binom{(k+\ell)/2}{p}} + \underbrace{\max\left(|L_1|, \frac{|L_1|^2}{2^\ell}\right)}_{|L_1|と|L_2|のマージ}$

- $\mathrm{WF} = T/P$

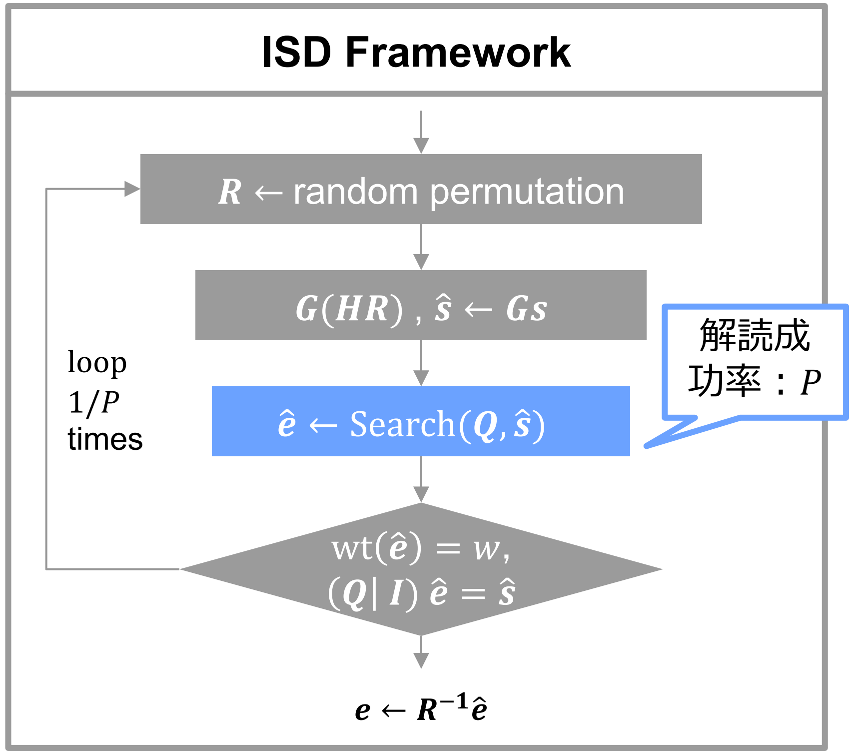


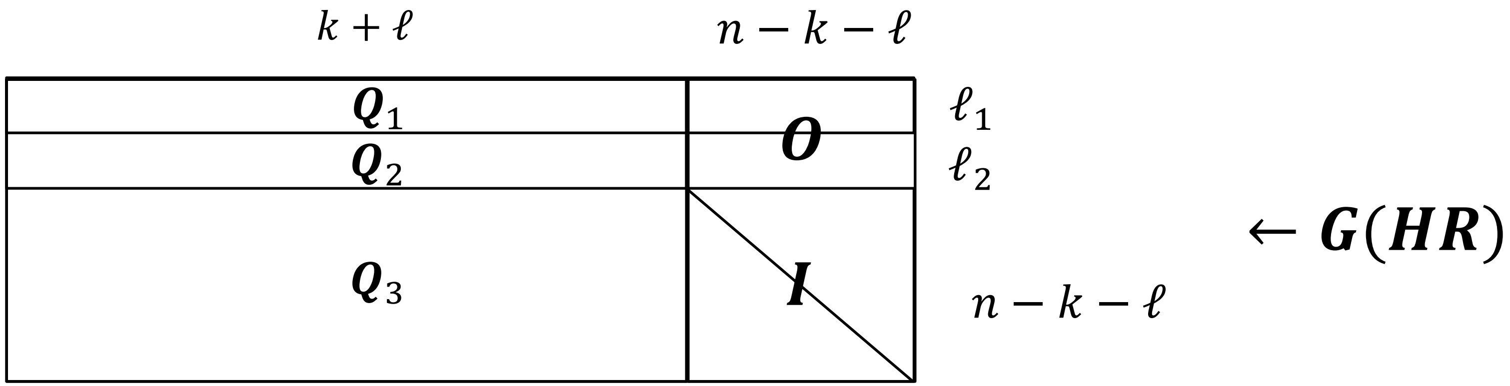
目次

May-Meurer-Thomae (MMT)

KDDI Research

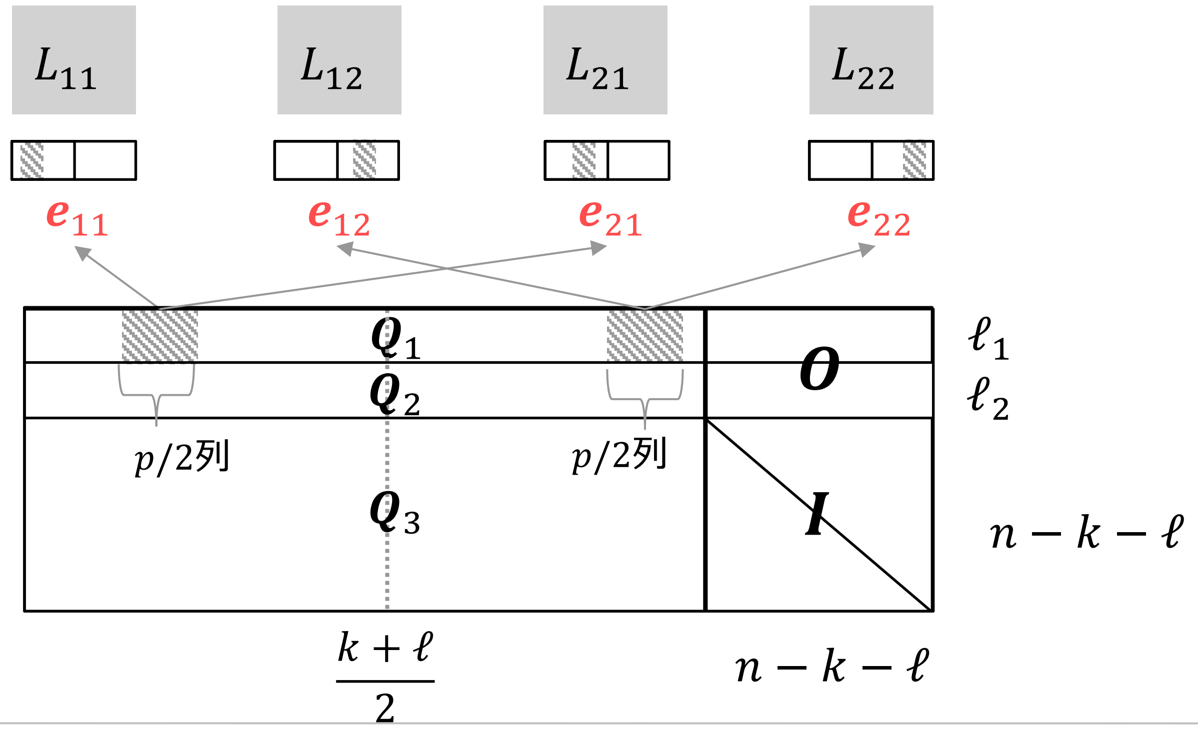
- 初期リストを4に増やしてマージを行う
- マージ時に分割表現(split representation)を利用
- $\boldsymbol{H}$は右図のように変換





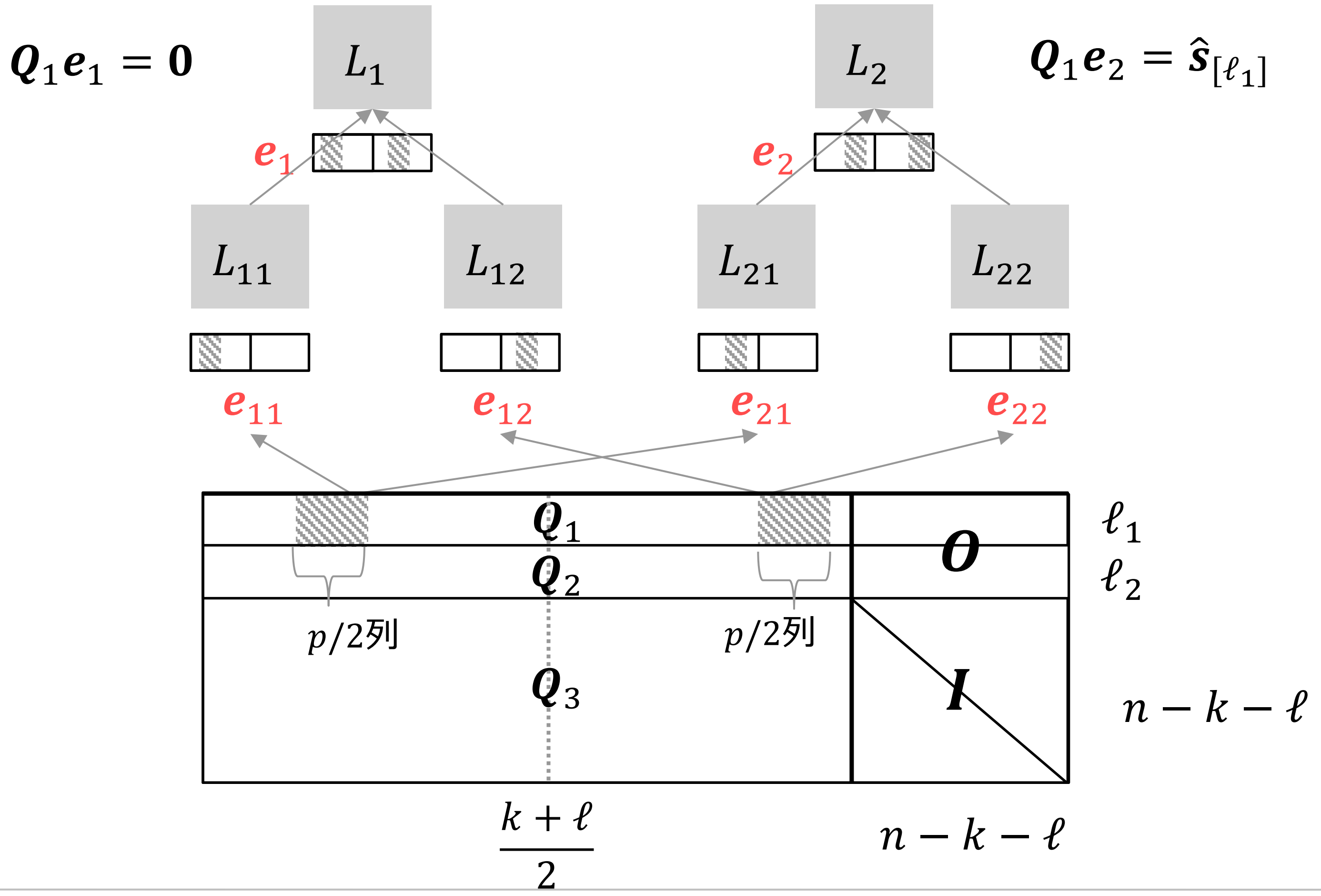
May-Meurer-Thomae (MMT)

- $\boldsymbol{Q}_1$の左右半分から$p/2$列の数え上げを行い、4つのリスト$L_{11}, L_{12}, L_{21}, L_{22}$を構築



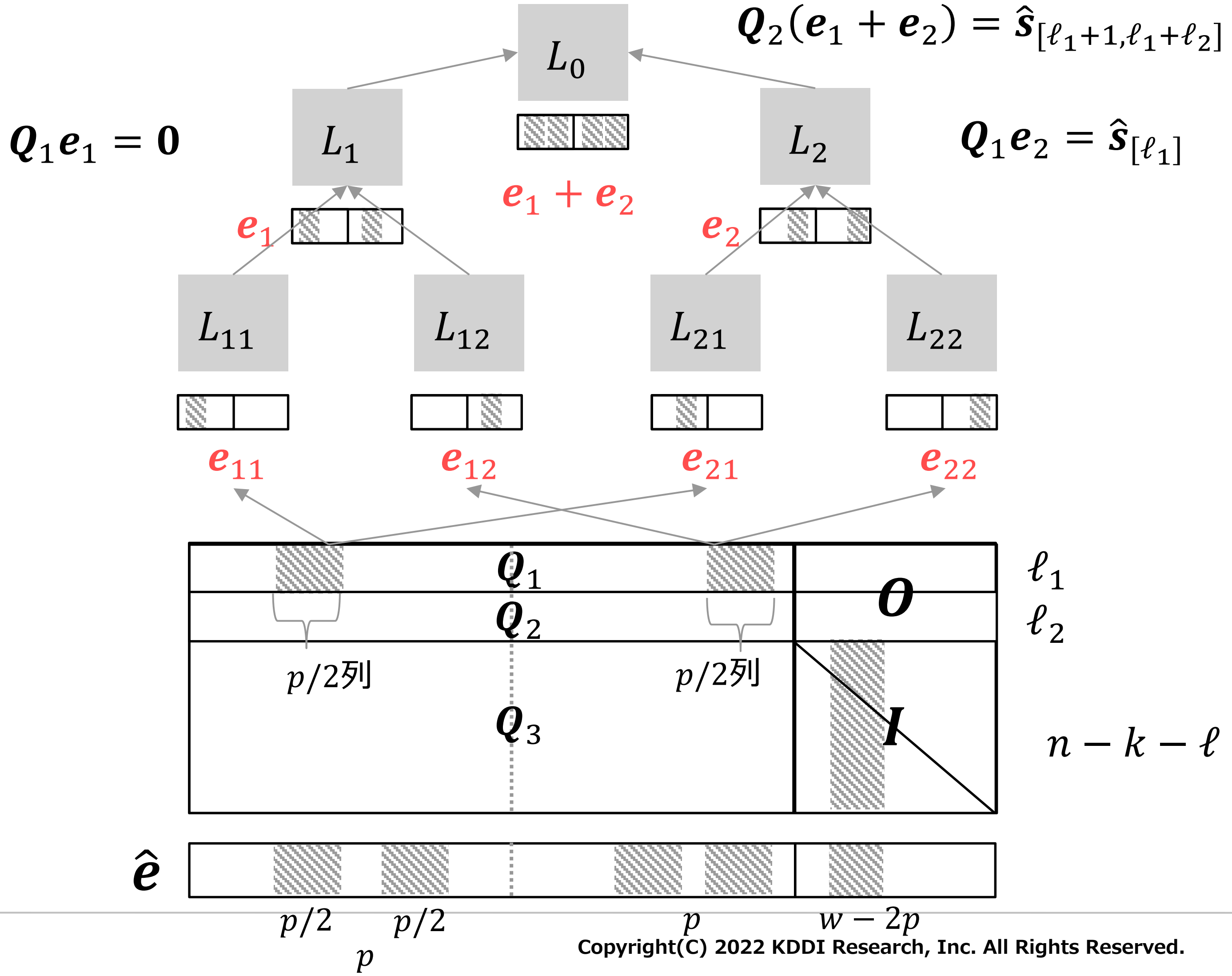
May-Meurer-Thomae (MMT)

- 4つのリストを$\boldsymbol{Q}_1\boldsymbol{e}_{11} = \boldsymbol{Q}_1\boldsymbol{e}_{12}$($\boldsymbol{Q}_1\boldsymbol{e}_{21} = \boldsymbol{Q}_1\boldsymbol{e}_{22} + \hat{\boldsymbol{s}}_{[\ell_1]}$)を満たすようにマージ



May-Meurer-Thomae (MMT)

■ 2つのリストを$\boldsymbol{Q}_2\boldsymbol{e}_1 = \boldsymbol{Q}_2\boldsymbol{e}_2 + \hat{\boldsymbol{s}}_{[\ell_1+1,\ell_1+\ell_2]}$を満たすようにマージ ➔ $\boldsymbol{Q}_3$の検証



May-Meurer-Thomae (MMT) のWF

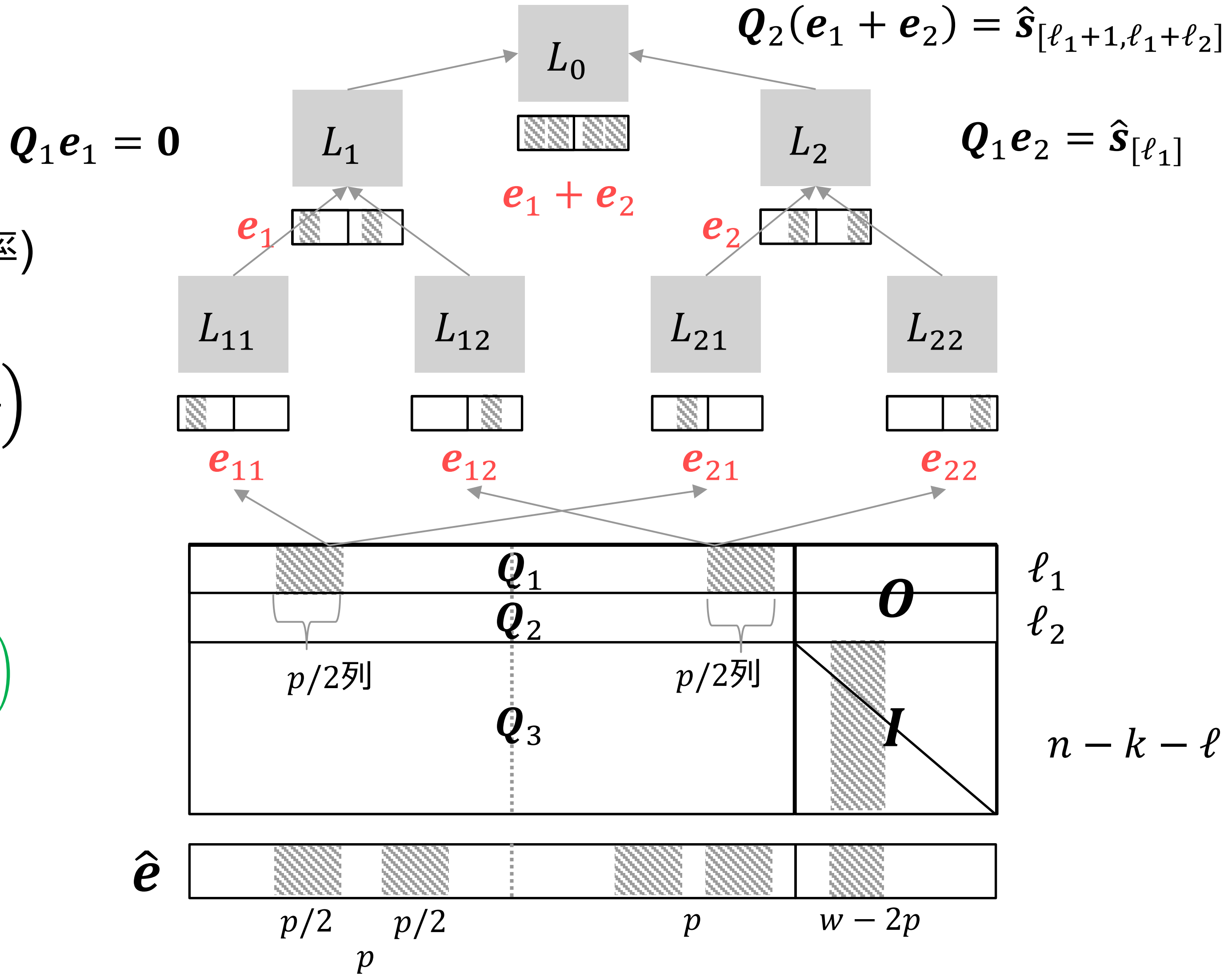
- $P = \dfrac{\binom{(k+\ell)/2}{p}^2 \binom{n-k-\ell}{w-2p}}{\binom{n}{w}}$

(w個の"1"が$\hat{e}$の区間に$p, p, w-2p$で分布する確率)

※厳密には異なる

- $T = n(n-k) + \underbrace{|L_{11}|}_{|L_{11}| = \binom{(k+\ell)/2}{p/2}} + \max\left(|L_{11}|, \dfrac{|L_{11}|^2}{2^{\ell_1}}\right) + \max\left(\underbrace{|L_1|}_{|L_1| = \max\left(1, \frac{|L_{11}|^2}{2^{\ell_1}}\right)}, \dfrac{|L_1|^2}{2^{\ell_2}}\right)$

- $\mathrm{WF} = T/P$



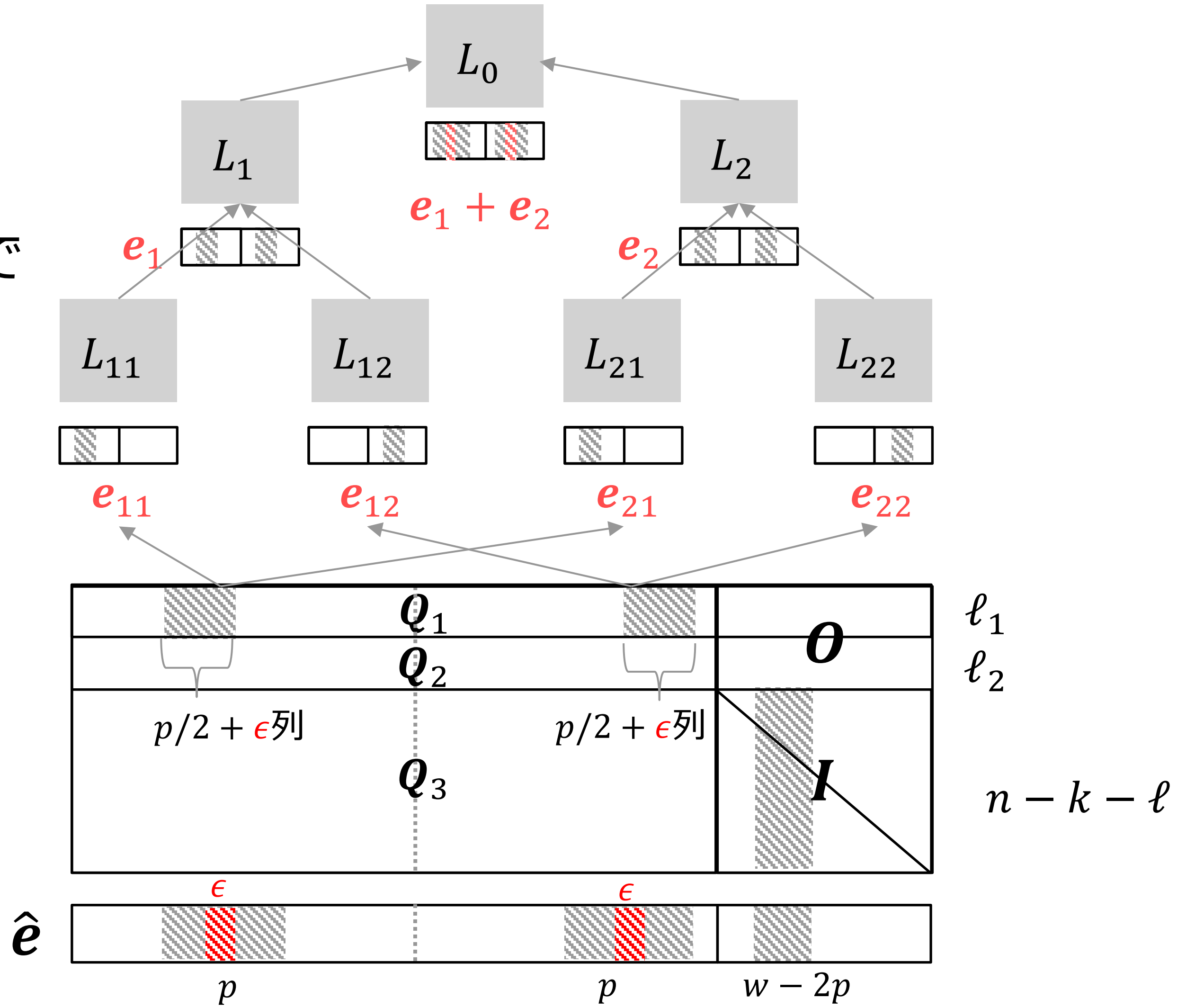
目次

Becker-Joux-May-Meurer (BJMM)

KDDI Research

- MMTをパラメータpに関して一般化
- 初期リストの重みに**追加重みϵ**を考慮
- 追加重みは最後のマージで打ち消される（$\boldsymbol{e}_1$と$\boldsymbol{e}_2$の同じ位置の"1"は$\hat{\boldsymbol{e}} = \boldsymbol{e}_1 + \boldsymbol{e}_2$で$1 + 1 = 0$となる）

- MMTと同じ成功確率P
- **分割表現がMMTより優れている**ため、問題によってはMMTより小さいWF



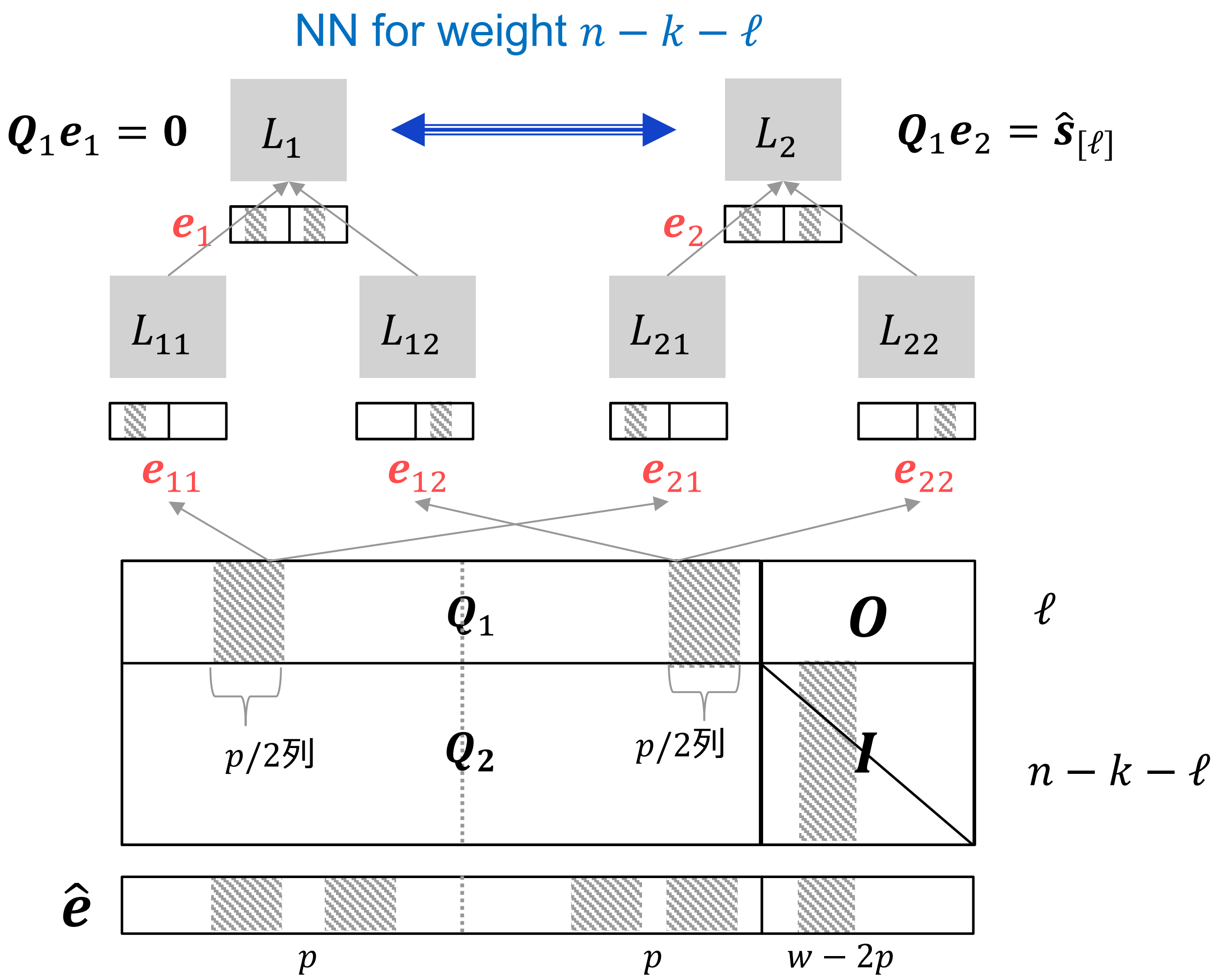
目次

May-Ozerov (MO)

- 深さ1のリストのマージの代わりに**最近傍法(NN: Nearest Neighbor)**を使う
- MMTの場合を考える
  - $\ell_1 = \ell, \ell_2 = 0$とおく
- $\boldsymbol{Q}_2\boldsymbol{e}_1$と$\boldsymbol{Q}_2\boldsymbol{e}_2 + \hat{\boldsymbol{s}}_{[\ell+1,n-k]}$に対して $\mathrm{wt}(\boldsymbol{Q}_2\boldsymbol{e}_1 + \boldsymbol{Q}_2\boldsymbol{e}_2 + \hat{\boldsymbol{s}}_{[\ell+1,n-k]}) = w - 2p$ となるペアをNNで探索

  ➔ NNの解がSDPの解

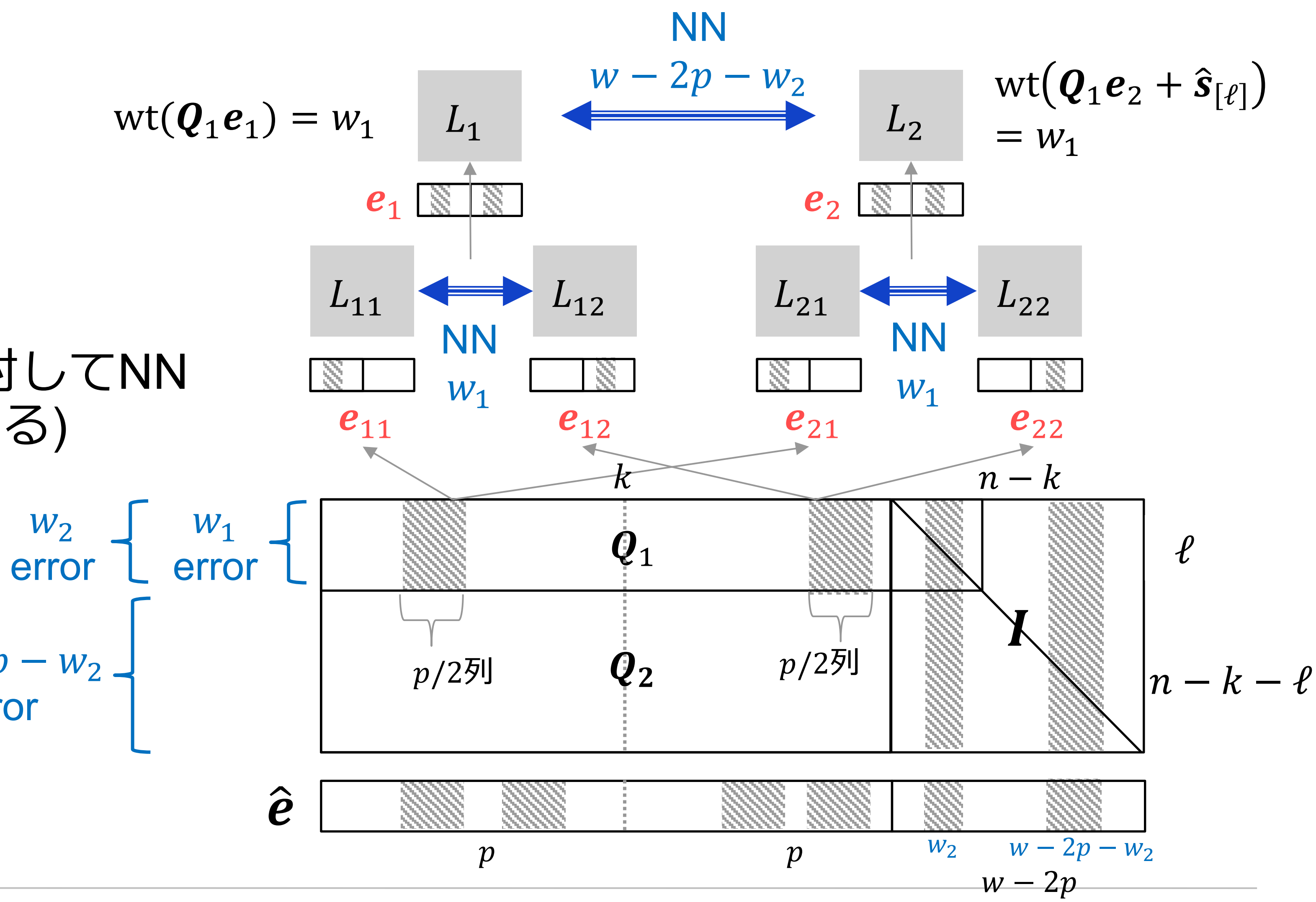
- PはMMTの場合と同じ
- Tは深さ1のリストのマージの計算量をNNの計算量で置き換えたもの



目次

Both-May (BM)

- **全てのの深さのリストのマージ**に最近傍法(Nearest Neighbor)を使う
- ここでは深さ2を考える
- 深さ2 → 1のマージ：
  $\boldsymbol{Q}_1$に対して重みw_1に対してNN
- 深さ1 → 0のマージ：
  $\boldsymbol{Q}_2$に対して重み$w-2p-w_2$に対してNN
  (同時に$\boldsymbol{Q}_1$の重み$w_1 \to w_2$を考える)
- $P = \dfrac{\binom{k/2}{p}^2\binom{\ell}{w_2}\binom{n-k-\ell}{w-2p-w_2}}{\binom{n}{w}}$
- Tはリストのマージの計算量をNNの計算量で置き換えたもの



目次

Syndrome Decoding Estimator (SDE)

KDDI Research

- 具体的なSDPインスタンス$SDP(n, k, w)$に対してISDの実計算量を推定するプログラム
- これまでにいくつかのSDEが提案 [EB22,BB+19,HS13,Pet10]
- 最近Esserらによって提案されたSDE [EB22]を使って**メモリ制限時**の計算量解析を実施

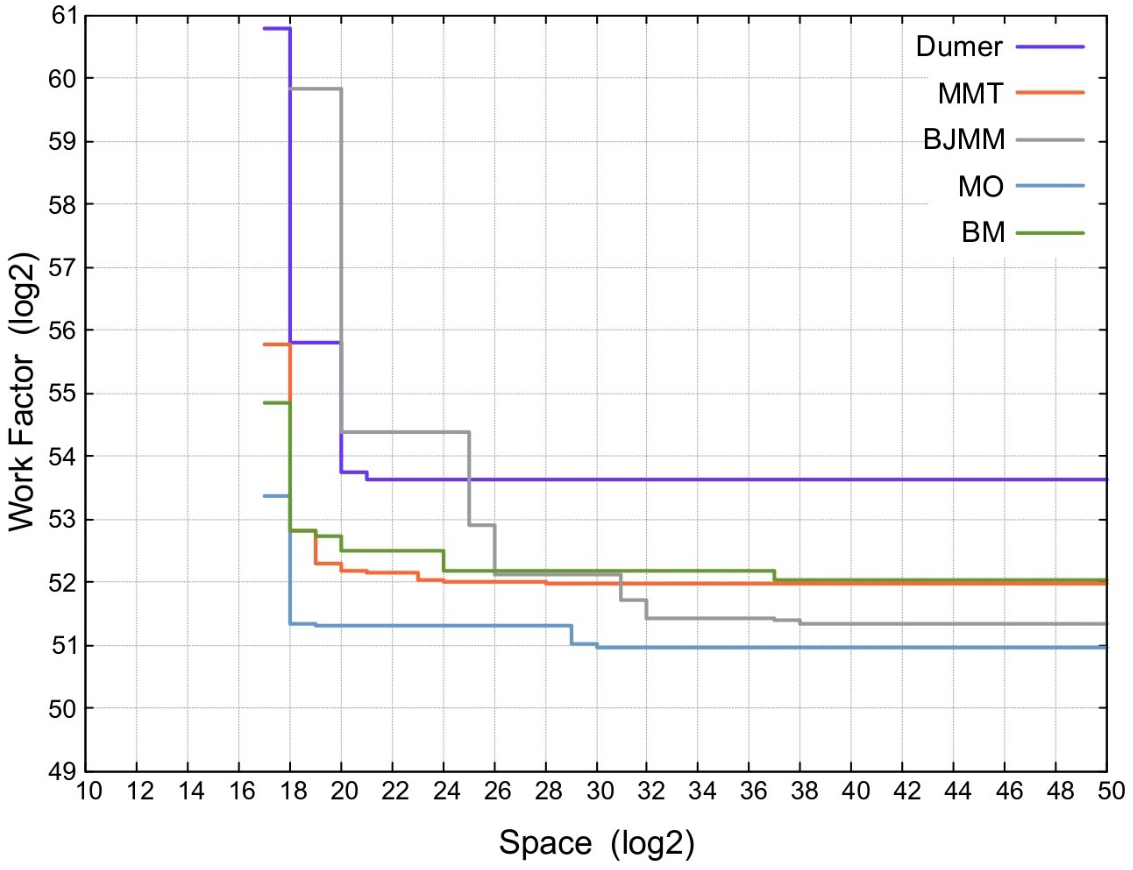
アルゴリズム（著者名）	略称	年度	漸近計算量	実計算量
Prange [Pra62]	–	1962	$2^{0.121n}$	?
Dumer [Dum91]	–	1991	$2^{0.117n}$	?
May-Meurer-Thomae [MMT11]	MMT	2011	$2^{0.112n}$	?
Becker-Joux-May-Meurer [BJMM12]	BJMM	2012	$2^{0.102n}$	?
May-Ozerov [MO15]	MO	2015	$2^{0.0953n}$	?
Both-May [BM18]	BM	2018	$2^{0.0885n}$	?

Syndrome Decoding Estimator (SDE)

KDDI Research

- 解析結果の一例（2021年当時未解読であったSDPインスタンス）
- 左：実計算量（メモリ制約あり）　右：漸近計算量
- 漸近計算量では不明であった各ISDの詳細なWFが判明 ➔省メモリではMMT・MOが高速

SDP(500,250,61) (難しさ：2^{53})



algorithm	漸近計算量
Prange	$2^{0.121n}$
Dumer	$2^{0.117n}$
MMT	$2^{0.112n}$
BJMM	$2^{0.102n}$
MO	$2^{0.0953n}$
BM	$2^{0.0885n}$

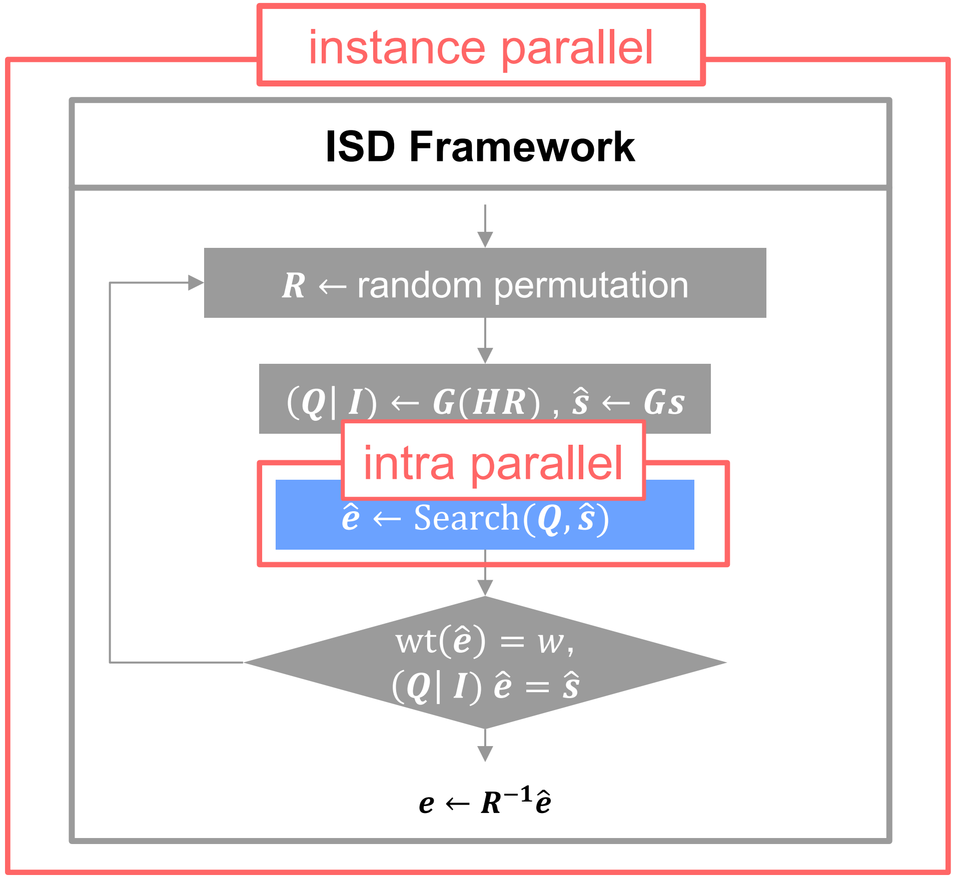
目次

KDDI Research

並列ISDアルゴリズム

KDDI Research

- 大規模なSDPを解読するには並列アルゴリズムが必須
- ISD全体を並列化させる手法(instance)と一部の内部処理を並列化(intra)する手法が存在



手法	HW	並列化手法	ISD
FPGA Stern Attack [HZP14]	FPGA	intra	Stern
Parallel MMT/BJMM [EMZ22]	CPU	instance	MMT/BJMM
Multiparallel Dumer [NFK21]	GPU	intra	Dumer
Multiparallel MMT [NFK22]	GPU	intra	MMT

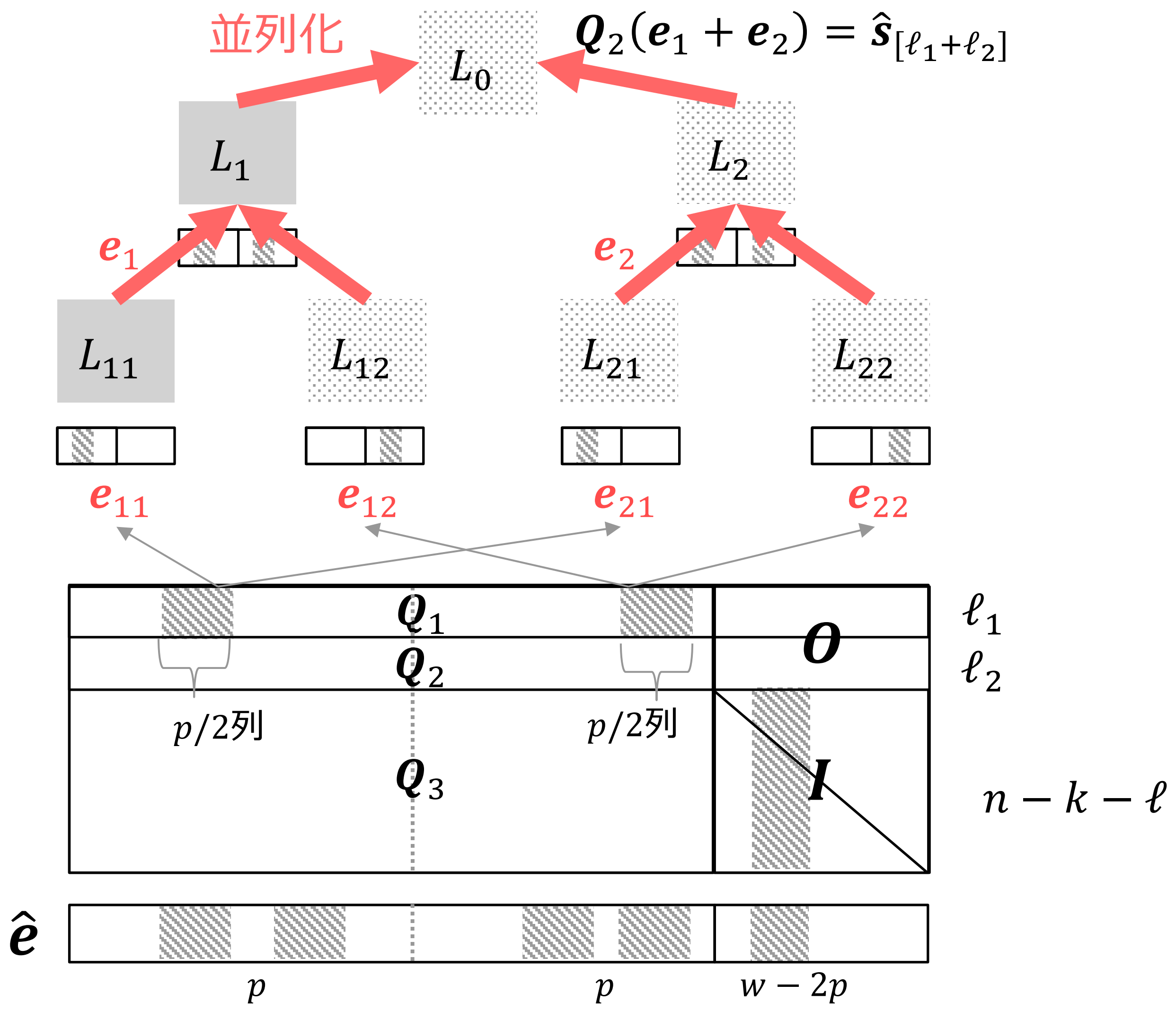
$WF = T/P$

- Instance parallel
  - T：小、P：小
- Intra parallel
  - T：大、P：大

Multiparallel MMT

KDDI Research

- 特徴：
  - 計算量の大きい**リストのマージ処理**を並列化
  - メモリ削減のためL_{11}, L_1のみを構築
  - マージ時に非同期処理（競合書き込み）を実施
- C++とCUDAで実装（公開予定）



Multiparallel MMTの実験結果

■ 実験環境

- GPU: NVIDIA Tesla V100
- CPU: AMD Ryzen 9 3900

■ 並列数の速度への影響

並列数 ($\|L_{11}\|$)	16	128	1024	11175 (max)
並列数 ($\|L_1\|$)	16	128	1024	262144 (max)
T (ms)	3005.55	402.37	57.14	1.63
速度比	1840.40	246.38	34.99	–

■ SDP(550,275,67)に対する計算時間の期待値・最適パラメータ

手法	期待値（年）	速度比
MMT (並列無し)	872.13	235
Multiparallel Dumer	21.75	5.9
Multiparallel MMT	3.69	–

p	ℓ	ℓ_1	ℓ_2
4	26	6	20
3	18	–	–
4	26	8	18

Multiparallel MMTの実験結果

■ SDP Challengeの解読結果：

$\mathrm{SDP}(n, k, w)$	Tesla V100 1台あたり期待値	#GPU	実際の解読時間
SDP(510,255,62)	153.6 days	4	24.7 days
SDP(530,260,65)	219.8 days	4	12.5 days
SDP(540,270,66)	461.5 days	22	79.44 days
SDP(550,275,67)	1350 days	4	13.03 days

- Multiparallel MMTのメモリ使用量：約300 [MB]

■ McEliece Challengeにおける解読時間の期待値：

$\mathrm{SDP}(n, k, w)$	Parallel MMT/BJMM [12] w/ 4 AMD EPYC 7742 CPU (512 threads)	Multiparallel MMT w/ 4 Tesla V100 GPU
SDP(1284,1028,24)	37.47 days	158.22 days

- Multiparallel MMTのメモリ使用量：約16.5 [GB]

符号暗号解読コンテストの結果 (2022.8.1現在)

KDDI Research

SDP Challenge

Home | Generic problems | NIST-like problems | Documentation | Contact

Syndrome Decoding Problem

Hall of Fame

Length	Weight	Authors	Algorithm	Date	Details
550	67	Shintaro Narisada, Kazuhide Fukushima, and Shinsaku Kiyomoto	MMT	2022-02-23	See details
540	66	Shintaro Narisada, Kazuhide Fukushima, and Shinsaku Kiyomoto	MMT	2022-02-01	See details
530	65	Shintaro Narisada, Kazuhide Fukushima, and Shinsaku Kiyomoto	MMT	2021-10-27	See details
510	61	Shintaro Narisada, Kazuhide Fukushima, and Shinsaku Kiyomoto	MMT	2021-09-19	See details
500	59	Greg Meyer	Dumer	2020-07-27	See details
490	59	Greg Meyer	Dumer	2020-08-01	See details
480	59	Greg Meyer	Dumer	2020-07-	See

符号暗号解読コンテストの結果 (2022.8.1現在)

Goppa McEliece Challenge

Home Generic problems NIST-like problems Documentation Contact

Syndrome Decoding in the Goppa-McEliece Setting

Hall of Fame

Length	Weight	Authors	Algorithm	Date	Details
1284	24	Andre Esser, Alex May and Floyd Zweydinger	MMT variant	2021-08-16	See details
1223	23	Andre Esser, Alex May and Floyd Zweydinger	BJMM/MMT variant	2021-05-10	See details
1161	22	Shintaro Narisada, Kazuhide Fukushima, and Shinsaku Kiyomoto	Dumer	2021-01-10	See details
1101	21	Anders Nilson	Multi threads Dumer4, Gregory Landais impl.	2020-08-14	See details
1041	19	Shintaro Narisada, Kazuhide Fukushima, and Shinsaku Kiyomoto	Dumer	2020-08-11	See details
982	20	Noémie Bossard	Mutltithreaded Dumer4, Gregory Landais original implementation	2020-07-02	See details
923	19	Valentin Vasseur	Dumer	2020-03-17	See details

目次

KDDI Research

量子Information Set Decoding

- 古典ISDに量子探索アルゴリズム(Grover探索、量子ウォーク探索)を適用
- 古典ISDの平方根くらいの計算量

古典アルゴリズム	漸近計算量	量子アルゴリズム	漸近計算量	手法
Prange [Pra62]	$2^{0.121n}$	量子Prange [Ber10]	$2^{0.0604n}$	Grover
Dumer [Dum91]	$2^{0.117n}$	量子Dumer [KT17]	$2^{0.0597n}$	Grover + QW
		量子Dumer + NN [Kir18]	$2^{0.0595n}$	Grover + QW
MMT [MMT11]	$2^{0.112n}$	量子MMT [KT17]	$2^{0.0590n}$	Grover + QW
BJMM [BJMM12]	$2^{0.102n}$	量子BJMM [KT17]	$2^{0.0587n}$	Grover + QW
MO [MO15]	$2^{0.0953n}$	–	–	–
BM [BM18]	$2^{0.0885n}$	–	–	–

* QW: 量子ウォーク探索 (Quantum Walk)

量子Information Set Decoding（量子回路構成・実装）

KDDI Research

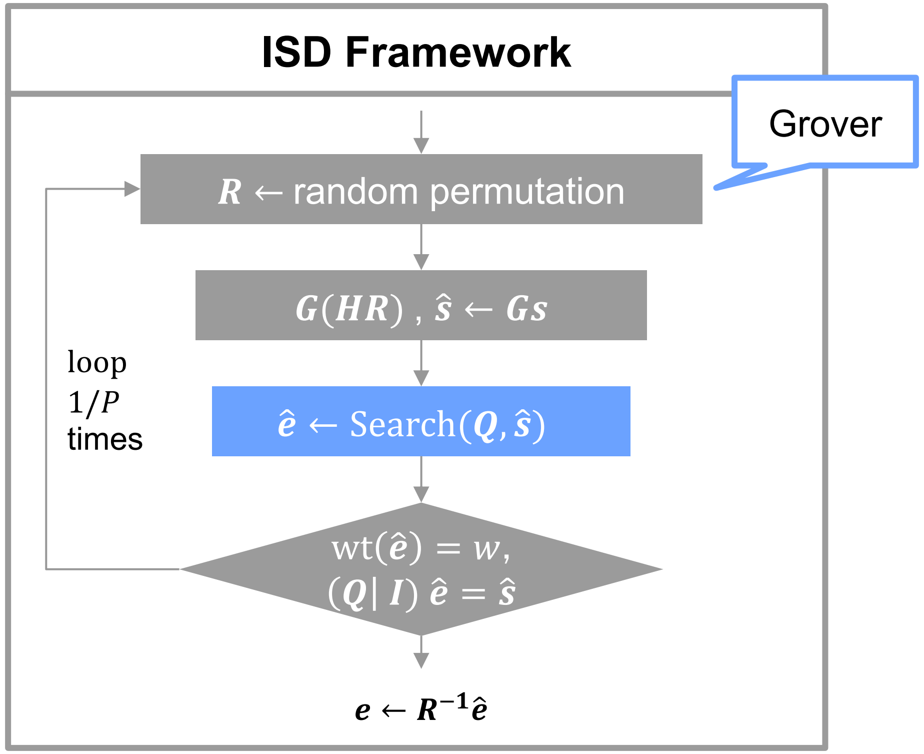
■ 最近は量子ISDの量子ゲート回路の構成を示す論文も登場している

量子アルゴリズム	漸近計算量	量子回路構成	手法	実装
量子Prange [Ber10]	$2^{0.0604n}$	[ECB+22,PBP21]	Grover, AA	Qibo [ECB+22]
量子Dumer [KT17]	$2^{0.0597n}$	[LL22*]	量子ウォーク	–
量子Dumer + NN [Kir18]	$2^{0.0595n}$	–	–	–
量子MMT [KT17]	$2^{0.0590n}$	–	–	–
量子BJMM [KT17]	$2^{0.0587n}$	–	–	–

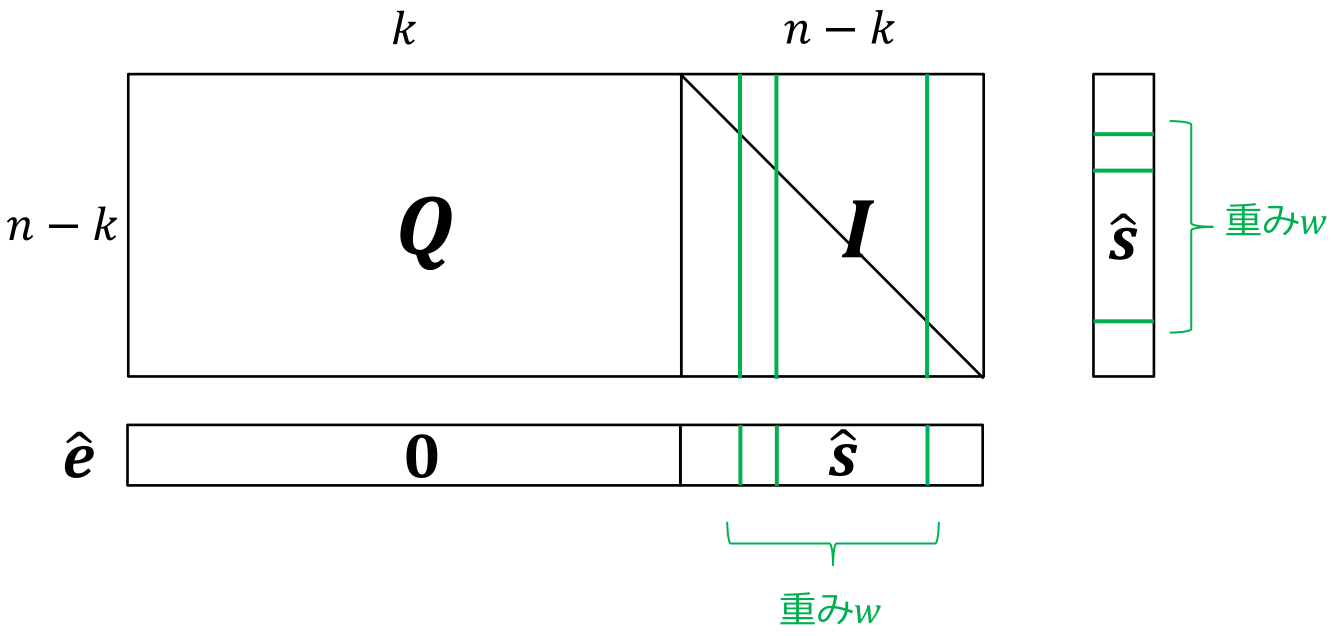
* 本来の量子Dumer [KT17]はGrover + 量子ウォークを行うが、
[LL22]は量子ウォークの部分のみの回路構成になっており、完全な量子Dumerの量子回路は未実現

量子Prange [Ber10]

- 一度の探索での解読成功率$P = \frac{\binom{n-k}{w}}{\binom{n}{w}}$ ➔ 期待ループ数：$1/P$
- $1/P$個のランダムな**permutationに対してGrover探索**を行うことで$1/\sqrt{P}$時間で確率的に解が求まる

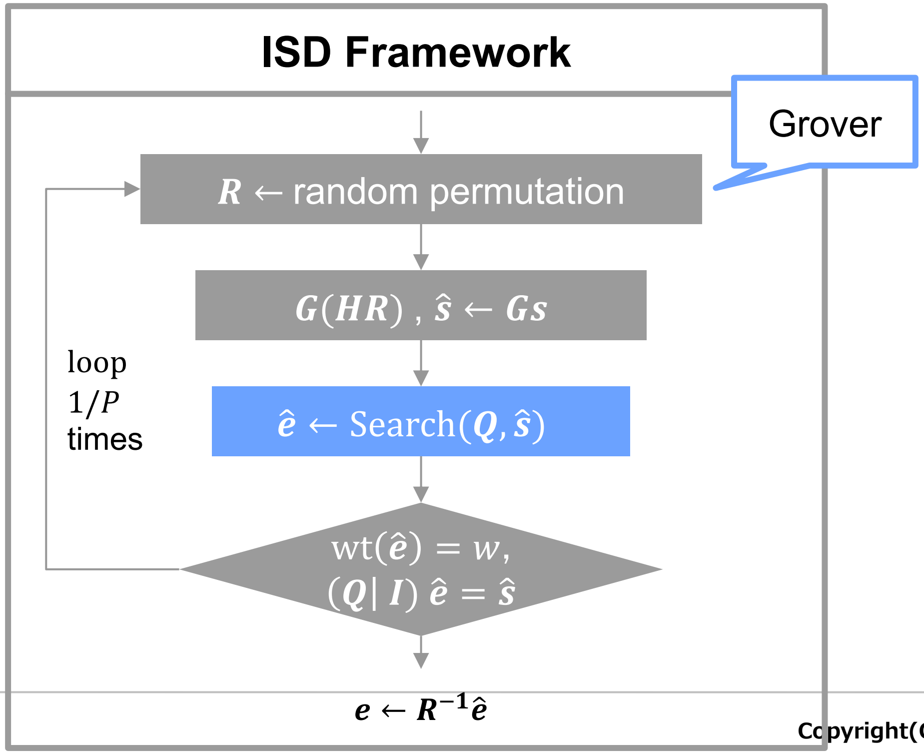


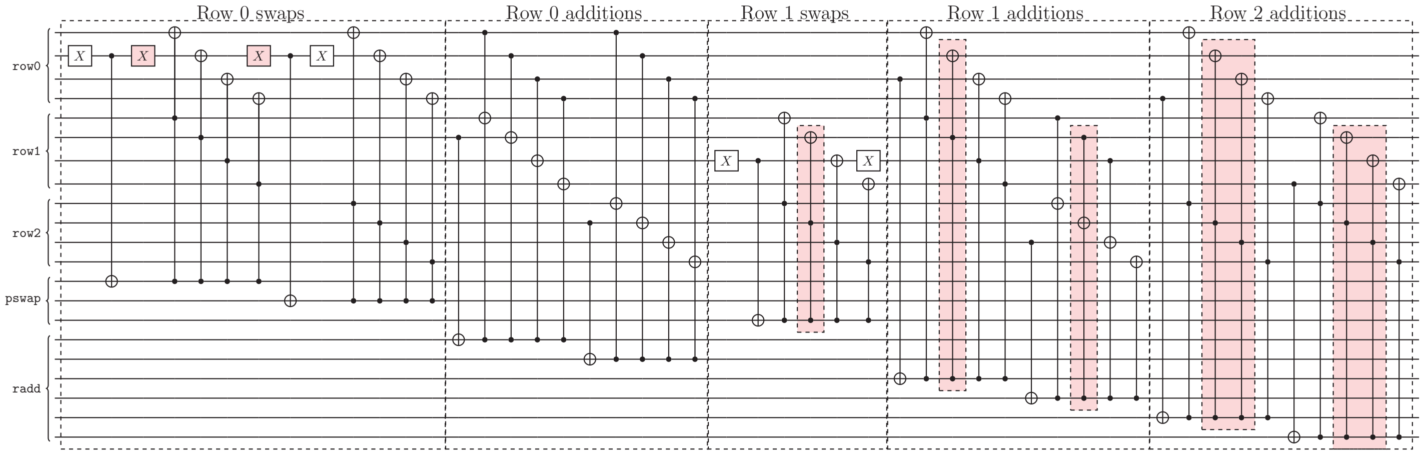
- 量子PrangeのWF $= n(n-k)\sqrt{\frac{\binom{n}{w}}{\binom{n-k}{w}}}$



量子Prangeの量子回路 [ECB+22,PBP21]

- Groverを行う前に所望の状態の重ね合わせを構築
- $1/P$個のランダムなpermutationの重ね合わせを作る代わりに、$\binom{n}{n-k}$の重ね合わせを用意
  - Dicke State [21]を使うと$O(n)$で$\binom{n}{n-k}$の重ね合わせを構築できる
- この重ね合わせに対して、量子ゲート上でGrover探索（permutation・ガウス消去法・ハミング重みの確認からなるoracle、diffusion）を行う

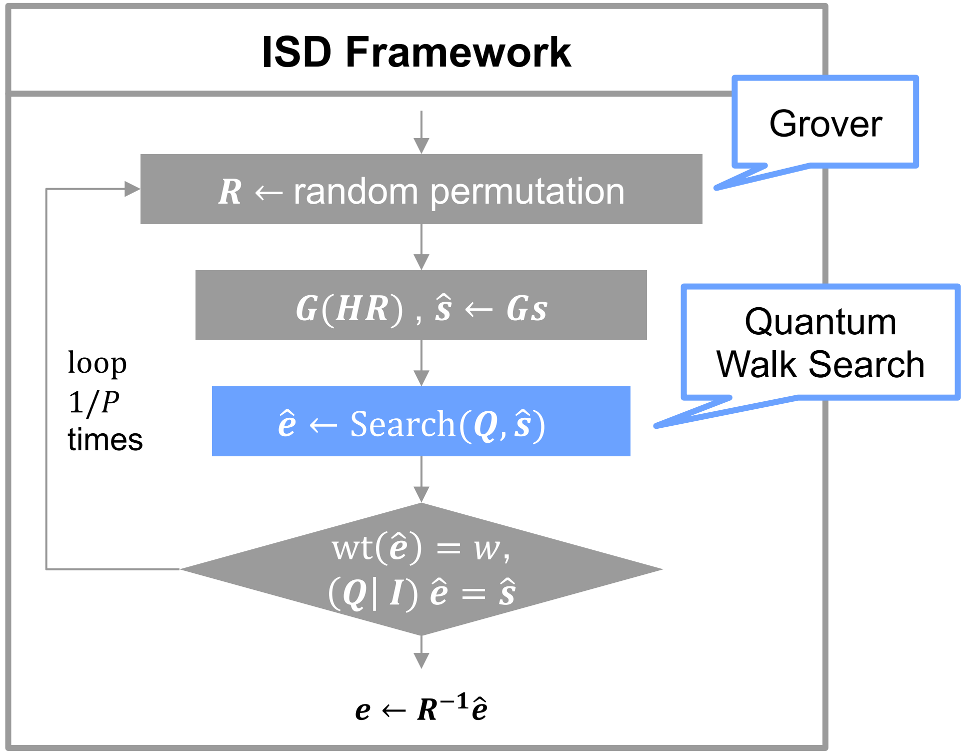


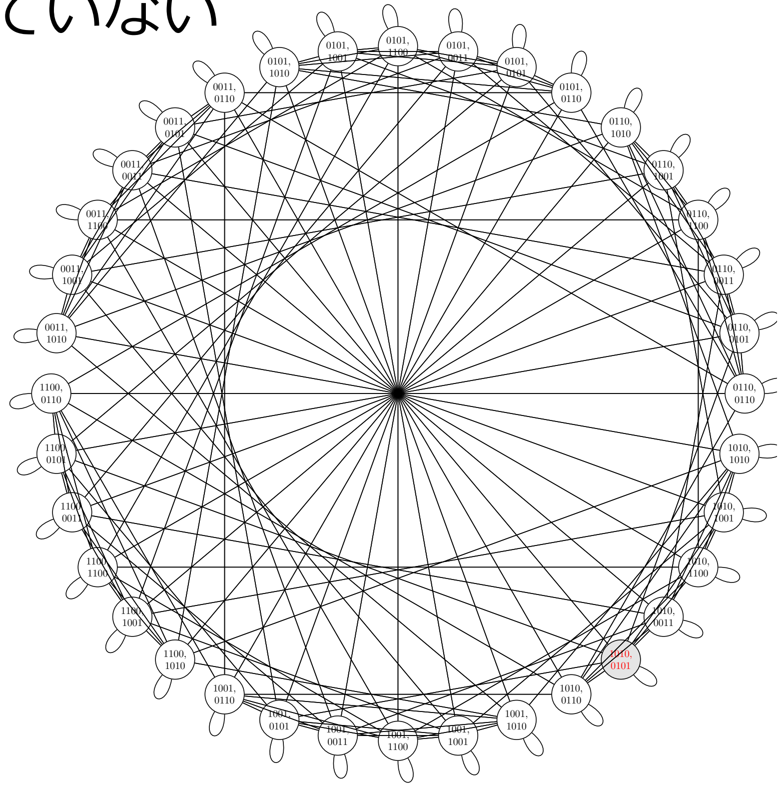


ガウス消去法の量子ゲート実装の一部

量子Dumer [KT17]・量子MMT/BJMM[Kir18]・量子MO/BM

- 各permutationに対してGrover探索を実行
- Grover探索の内部処理として量子ウォーク探索を実施
- 理論検討のみで実装・量子回路に関する研究は知られていない



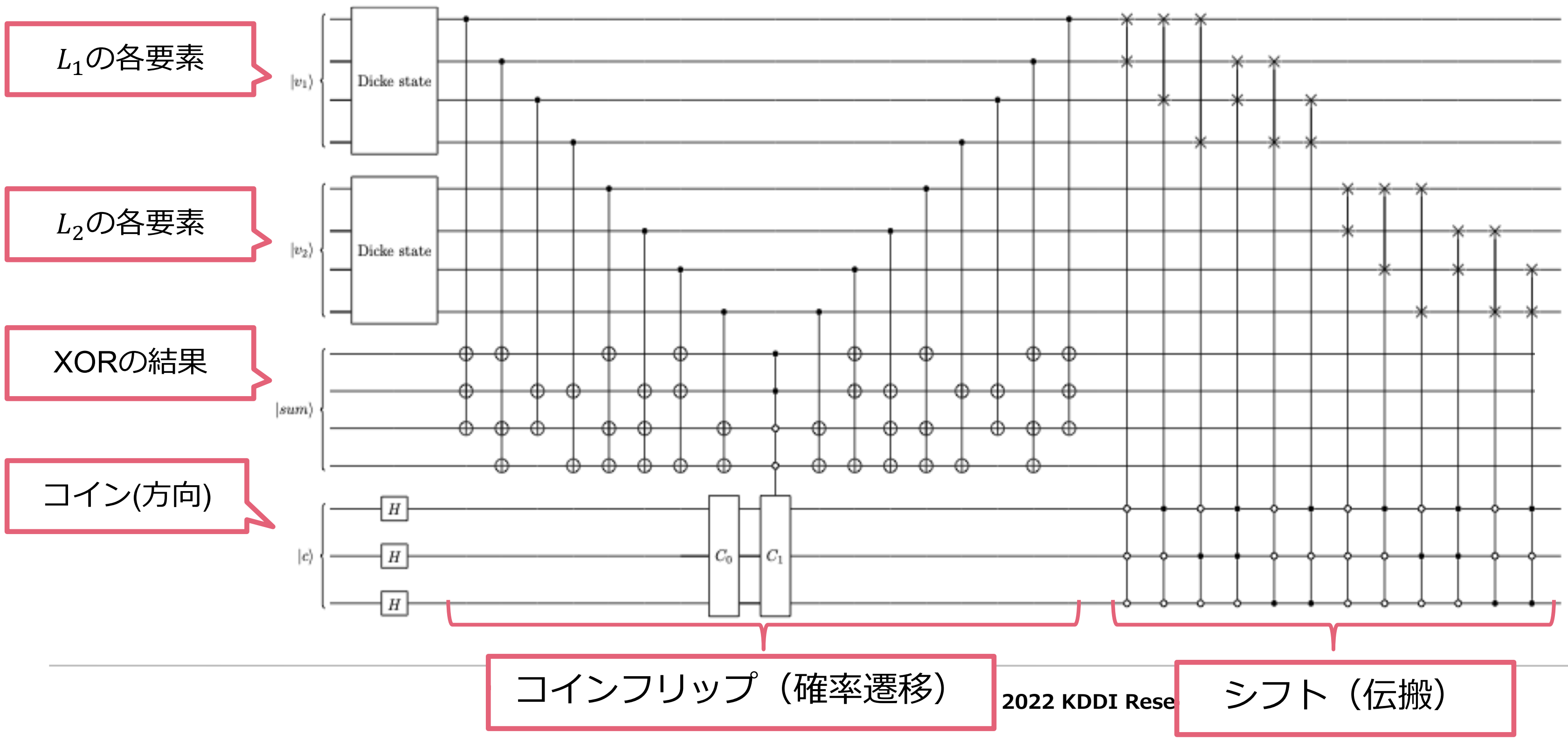


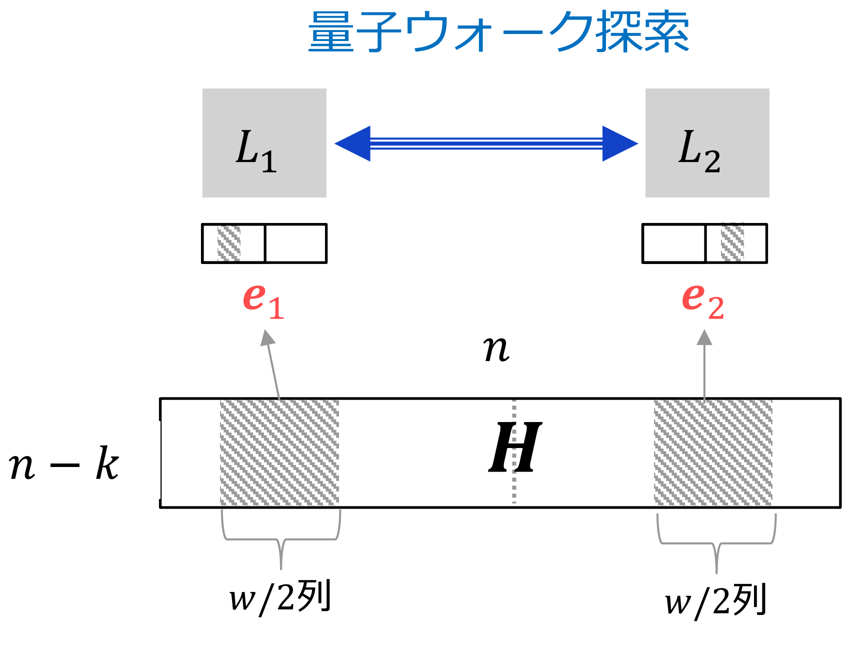
量子Dumerにおける量子ウォーク探索
（Johnsonグラフの直積グラフ上での探索）

量子Dumerの部分的な量子回路 [LL22]（量子BirthdayDecode）

- DumerのリストのマージBirthdayDecode)の部分を量子ウォーク探索で求解
- 測定すると1要素しか得られないので単体ではリストのマージには使えない
- permutation /ガウス消去法を行わず直接$\boldsymbol{H}$をBirthdayDecodeすることはできる（下図）
  - $\boldsymbol{s} = 1100$に対するBirthday Decodeの量子ウォーク回路（1 step）

    $\boldsymbol{H} = [1110,1011,0110,0101,1011,0111,1101,0011]$





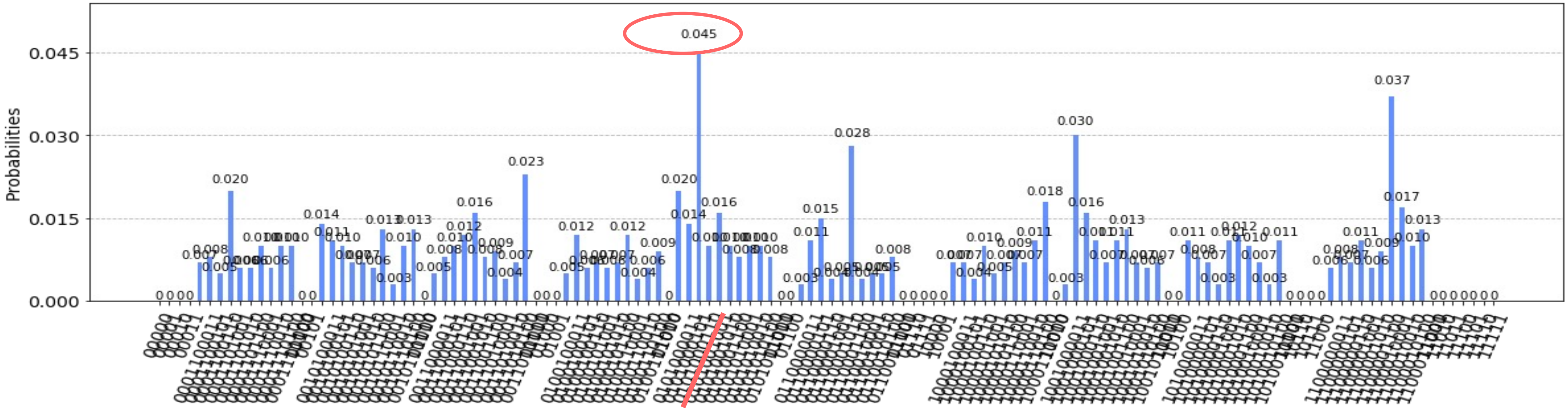
$C_0 = C_1 = 2|s\rangle\langle s| - I$ (diffuser)

量子BirthdayDecodeの検証実験

- Qiskitで先ほどの量子ウォーク探索を実装
- 量子Dumerではなく量子BirthdayDecode (permutation,ガウス消去法無し)
- シミュレータ上でDecoding Challengeの最小インスタンスSDP(10,5,4)の解読に成功

$$\boldsymbol{s} = 01001,\ \boldsymbol{H} = [10000,01000,00100,00010,00001,00101,11100,10110,11001,00001],\ w = 4$$

$$\boldsymbol{e}_1 = 0110000000 \quad \boldsymbol{e}_2 = 0000001010, \qquad \boldsymbol{e} = \boldsymbol{e}_1 + \boldsymbol{e}_2 = 0110001010$$



まとめ・今後の課題

- シンドローム復号問題 (SDP)と解読アルゴリズム (ISD)を紹介
- ISDの高速実装としては**多並列アルゴリズム**（CPU・GPU・FPGA）が主流
- ISDの**実計算量解析**に関する研究もなされている
- 量子ISDについては近年は**量子回路実装**が盛ん

- 今後の課題：
  - MMT以降の古典ISDアルゴリズムのGPU実装・FPGA実装・ASIC実装等
  - より漸近計算量の小さい古典ISD・量子ISDの研究開発
  - Grover + 量子ウォーク探索に基づく量子ISDの量子回路実装・回路計算量に関する研究
  - バイナリ以外 (ternary, cyclic等)のSDPに対する高速解読アルゴリズムの開発 ... etc.

ご清聴ありがとうございました

参考文献

- [Pra62] E. Prange, “The use of information sets in decoding cyclic codes,”1962.
- [Dum91] I. Dumer, “On minimum distance decoding of linear codes,”1991.
- [MMT11] A. May, A. Meurer, and E. Thomae, “Decoding random linear codes in ˜O (20.054n) ,” 2011.
- [BJMM12] A. Becker, A. Joux, A. May, and A. Meurer, “Decoding random binary linear codes in 2n/20: How 1 + 1 = 0 improves information set decoding,”2012
- [MO15] A. May and I. Ozerov, “On computing nearest neighbors with applications to decoding of binary linear codes,” 2015
- [BM18] L. Both and A. May, “Decoding linear codes with high error rate and its impact for LPN security,” 2018
- [EB22] A. Esser and E. Bellini, “Syndrome decoding estimator,” 2022.
- [BBC+19] M. Baldi, A. Barenghi, F. Chiaraluce, G. Pelosi, and P. Santini, “A finite regime analysis of information set decoding algorithms,” 2019.
- [HS13] Y. Hamdaoui and N. Sendrier, “A non asymptotic analysis of information set decoding.,” 2013.
- [Pet10] C. Peters, “Information-set decoding for linear codes over F q,” 2010.
- [HZP14] S. Heyse, R. Zimmermann, and C. Paar, “Attacking code-based cryptosystems with information set decoding using special-purpose hardware,” 2014
- [EMZ22] A. Esser, A. May, and F. Zweydinger, “McEliece needs a break – solving McEliece-1284 and Quasi-Cyclic-2918 with modern ISD,” 2022
- [NFK21] S. Narisada, K. Fukushima, and S. Kiyomoto, “Fast GPU implementation of dumer’s algorithm solving the syndrome decoding problem,” 2021.
- [NFK22] S. Narisada, K. Fukushima, and S. Kiyomoto, “Multiparallel MMT : Faster ISD Algorithm Solving High-Dimensional Syndrome Decoding Problem,” 2022.
- [Ber10] D. J. Bernstein, “Grover vs. McEliece,” 2010
- [KT17] G. Kachigar, and J. P. Tillich, “Quantum Information Set Decoding Algorithms,” 2017
- [Kir18] E. Kirshanova, “Improved Quantum Information Set Decoding,” 2018
- [ECB+22] A. Esser, S. R. Calderer, E. Bellini, J. I. Latorre, and M. Manzano, “Hybrid Decoding -- Classical-Quantum Trade-Offs for Information Set Decoding,” 2021
- [PBP21] S. Perriello, A. Barenghi and G. Pelosi, “A Complete Quantum Circuit to Solve the Information Set Decoding Problem,” 2021
- [LL22] G. Lancellotti, and M. Lodi, “Design of a Quantum Circuit for Quantum Random Walks on Johnson Graphs,” 2022
- [BE19] A. Bärtschi, and S. Eidenbenz, “Deterministic Preparation of Dicke States,” 2019
- [CD22] W. Castryck, and T. Decru, “An Efficient Key Recovery Attack on SIDH (preliminary version)”, 2022

漸近計算量

■ Stirling近似($\binom{n}{k} \sim 2^{nH\left(\frac{k}{n}\right)}$)を用いて導出された各ISDの最悪時計算量
($H(x)$は二値エントロピー関数)

漸近計算量

$$\max_{k} 2^{\alpha(k)n} \text{ for all } k = cn,\ 0 \leq c \leq 1$$

■ Full Distance Decodingと呼ばれる$\binom{n}{w} \sim 2^{n-k}$を満たすSDPに対して評価されることが多い
- 本発表でも特に断らない限りFull Distance Decodingを考える

■ Full Distance Decodingでは、重み$w = H^{-1}(1-k)$で固定

■ 手計算で漸近計算量を求めるのは大変なので、実際には計算機を用いて$\max_{k} 2^{\alpha(k)n}$の近似値を求める（cを刻み幅0.01でforループさせたり最適化ソルバを使う)

HW構成

- ローカル：デスクトップPCに無理矢理Tesla V100を搭載
- クラウド：AWS P3インスタンス (p3.2xlarge)

パフォーマンス比較

	ローカル	クラウド
解読性能	2.22 [ms]	2.98 [ms]

暗号解読用PC

- 解読実験はローカルx4 or クラウドx22で実施
- パーミュテーションの重複が発生しないよう調整