Algorithms for Bichromatic Closest Pairs Problem and application to Code-based Cryptography

M. Hamdad

April, 2025

The Bichromatic Closest Pairs Problem and application to cryptanalysis

- Boolean context

The Bichromatic Closest Pairs Problem and application to cryptanalysis

- Boolean context
- Syndrome Decoding Problem (SDP) :
  Find x such that $Hx = s$ with $wt(x) \leq w$ (NP-hard)

- Boolean context
- Syndrome Decoding Problem (SDP) :
  Find x such that $Hx = s$ with $wt(x) \leq w$ (NP-hard)
- Solve SDP faster
  $\implies$ improve McEliece cryptosystem's cryptanalysis

The Bichromatic Closest Pairs Problem and application to cryptanalysis

- Boolean context
- Syndrome Decoding Problem (SDP) :
  Find x such that $Hx = s$ with $wt(x) \leq w$ (NP-hard)
- Solve SDP faster
  $\Longrightarrow$ improve McEliece cryptosystem's cryptanalysis
- The best-known algorithms use in a crucial way a subroutine that solves $BCPP$

Bichromatic Closest Pairs Problem in $\mathbb{F}_2^m$

L

Bit n°	0	1	2	3	4	5	6
Line 0	1	1	1	1	1	0	0
Line 1	1	0	0	0	0	0	1
Line 2	0	1	1	1	1	0	0
Line 3	1	0	0	1	0	0	0
Line 4	0	1	1	0	1	0	1
Line 5	0	0	0	1	0	0	0
Line 6	1	0	0	0	1	0	0
Line 7	0	0	0	0	0	1	1

C

R

Bit n°	0	1	2	3	4	5	6
Line 0	1	1	1	1	0	1	0
Line 1	1	1	1	0	1	0	1
Line 2	1	1	1	0	0	0	0
Line 3	1	0	0	1	1	1	1
Line 4	1	0	0	0	1	1	1
Line 5	0	1	1	1	0	1	0
Line 6	0	1	1	0	0	1	1
Line 7	0	0	0	1	1	1	1

$N=2^{\lambda m}$

m

Goal: Find all $C = (x, y) \in L \times R$ such that $wt(x + y) \leq \gamma m$

Here $\gamma m = 1$

Bichromatic Closest Pairs Problem in $\mathbb{F}_2^m$

- The model : $\mathcal{M}_{Alea}\left(\mathbb{F}_2, 2^{\lambda m}\right)$
- The expected number of solutions : $E = \Theta\left(m^{-\frac{1}{2}} 2^{m(2\lambda + \mathrm{H}(\gamma) - 1)}\right)$[1]
- Asymptotically, the best algorithms : Carrier [Car20], Esser et. al [EKZ21] and May-Ozerov [MO15]

[1] H : Binary entropy function

Contribution [BDH25]

The May-Ozerov algorithm is galactic.

- In practice : Syndrome Decoding Estimator [EB21] $\implies$ The projection method
- **The Crossover Point:**
  The smallest code length n such that Decode(MO) can be faster than Decode(Projection).

R	D	n	MO $\log_2 N$	$\log_2 T$
0.5	0.11	533502	8121	29566
0.8	0.03	1874400	22282	63487

Table: Instance characteristics at the crossover point.

The projection method

Probability that 2 bit strings we search coincide on k columns: $p = \frac{\binom{(1-\gamma)m}{k}}{\binom{m}{k}}$

The projection method

Probability that 2 bit strings we search coincide on k columns: $p = \frac{\binom{(1-\gamma)m}{k}}{\binom{m}{k}}$

The algorithm

- Pick k columns randomly
- Sort the 2 lists in lexicographical order according to the selected columns
- Compare all pairs of bit strings that coincide on the k columns
- Repeat $\simeq \frac{1}{p}$ times

$k = 2$, drawn column numbers $= \{0, 2\}$

L sorted

Bit n°	2	0	1	3	4	5	6
Line 0	0	0	0	1	0	0	0
Line 1	0	0	0	0	0	1	1
Line 2	0	1	0	0	0	0	1
Line 3	0	1	0	1	0	0	0
Line 4	0	1	0	0	1	0	0
Line 5	1	0	1	1	1	0	0
Line 6	1	0	1	0	1	0	1
Line 7	1	1	1	1	1	0	0

C

R sorted

Bit n°	2	0	1	3	4	5	6
Line 0	0	0	0	1	1	1	1
Line 1	0	1	0	1	1	1	1
Line 2	0	1	0	0	1	1	1
Line 3	1	0	1	1	0	1	0
Line 4	1	0	1	0	0	1	1
Line 5	1	1	1	1	0	1	0
Line 6	1	1	1	0	1	0	1
Line 7	1	1	1	0	0	0	0

$k = 2$, drawn column numbers $= \{1, 4\}$

L sorted

Bit n°	4	1	0	2	3	5	6
Line 0	0	0	1	0	0	0	1
Line 1	0	0	1	0	1	0	0
Line 2	0	0	0	0	0	1	1
Line 3	0	0	0	0	1	0	0
Line 4	1	0	1	0	0	0	0
Line 5	1	1	1	1	1	0	0
Line 6	1	1	0	1	1	0	0
Line 7	1	1	0	1	0	0	1

C

R sorted

Bit n°	4	1	0	2	3	5	6
Line 0	0	1	1	1	1	1	0
Line 1	0	1	1	1	0	0	1
Line 2	0	1	0	1	1	1	0
Line 3	0	1	0	1	0	1	1
Line 4	1	0	1	0	1	1	1
Line 5	1	0	1	0	0	1	1
Line 6	1	0	0	0	1	1	1
Line 7	1	1	1	1	0	0	1

The expected number of solutions : E

Complexity

$$C_{Proj} = O\left(\left(N + \frac{N^2}{2^k}\right)\frac{1}{p}\right)$$

The expected number of solutions : E

Complexity

$$C_{Proj} = O\left(\left(N + \frac{N^2}{2^k}\right)\frac{1}{p}\right)$$

If $\lambda \leq \bar{\gamma}$, we take $k = \lambda m$ else $k = (1-2\gamma)m$ then :

$$C_{Proj} = \begin{cases} O\left(2^{m(\lambda + h(\lambda,\gamma))}\right) & \textit{for } \lambda < \bar{\gamma} \\ O\left(\sqrt{m}E\right) & \textit{for } \lambda \geq \bar{\gamma} \end{cases}$$

with

$$\bar{\gamma} = (1-2\gamma) \text{ and } h(\lambda,\gamma) = \mathrm{H}(\lambda) - (1-\gamma)\mathrm{H}\left(\frac{\lambda}{1-\gamma}\right)$$

Carrier [Car20], Esser et. al [EKZ21] and May-Ozerov [MO15] are similar.

A high-level description

- m columns are partitioned into t strips.
- The vectors of both lists are filtered strip after strip.
- Filtered across their m coordinates, the lists are small enough for an exhaustive search to find solution pairs

The May-Ozerov algorithm

Three core phases

1. Computing the parameters of the algorithm.
2. Double rerandomization.
3. Recursive search for solutions in a tree of depth t.

Algorithm 1 The MO algorithm

function $\mathrm{MO}(\ L, R, \lambda, \gamma, m, t, \epsilon\ )$

  $y(\gamma, \lambda) \leftarrow (1-\gamma)\left(1 - H\left(\frac{H^{-1}(1-\lambda)-\gamma/2}{1-\gamma}\right)\right)$

  $\alpha_1 \leftarrow (y(\gamma,\lambda) - \lambda + \epsilon/2)/y(\gamma,\lambda)$

  for $2 \leq j \leq t$ **do**

    $\alpha_j \leftarrow \frac{\lambda}{y(\gamma,\lambda)}\alpha_{j-1}$ ▷ Divide the lists into t strips of $\alpha_j m$ indices

  for $f_1(m)$ uniformly random permutation π of $\{1, \ldots m\}$ **do**

    for $f_2(m)$ times **do**

      $r = (r_1, \ldots, r_t) \leftarrow \left(\textsc{Random}(\mathbb{F}_2^{\alpha_j m})\right)_{j=1}^{t}$ s.t. $wt(r_j) = \alpha_j \frac{m}{2}$

      $\bar{L} \leftarrow \pi(L) + r$

      $\bar{R} \leftarrow \pi(R) + r$

      Remove from $\bar{L}$ and $\bar{R}$ all vectors that are not of weight $\alpha_j \frac{m}{2}$ on the j-th strip

      $C \leftarrow \textsc{RecursiveMO}(\bar{L}, \bar{R}, m, t, \epsilon, \gamma, \lambda, (\alpha)_{j=1}^{t}, 1)$

      if $C \neq \perp$ **then return** C

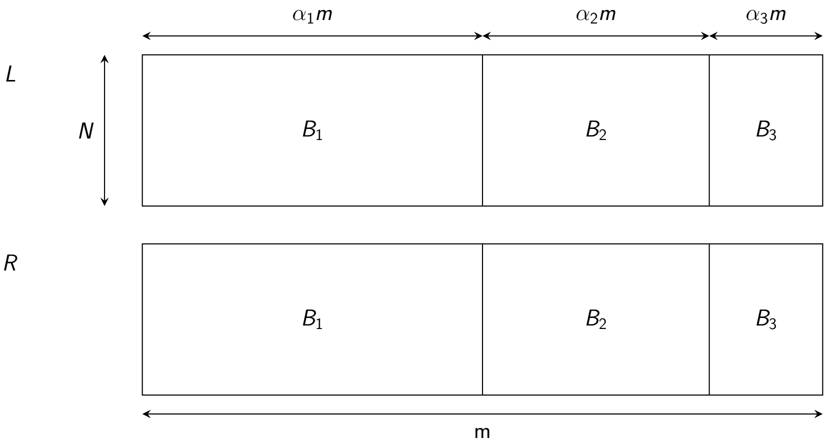
  return $\perp$

Parameter setup:

- $y(\gamma, \lambda)$
- Divide the lists into t strips $B_1, \ldots, B_t$ containing $\alpha_1 m, \ldots, \alpha_t m$ indices respectively, such that:

$$\sum_{j=1}^{t} \alpha_j m = m$$

$t = 3$



Algorithm 1 The MO algorithm

function $\mathrm{MO}(\ L, R, \lambda, \gamma, m, t, \epsilon\ )$

  $y(\gamma,\lambda) \leftarrow (1-\gamma)\left(1 - H\left(\frac{H^{-1}(1-\lambda)-\gamma/2}{1-\gamma}\right)\right)$

  $\alpha_1 \leftarrow (y(\gamma,\lambda) - \lambda + \epsilon/2)/y(\gamma,\lambda)$

  for $2 \le j \le t$ **do**

    $\alpha_j \leftarrow \frac{\lambda}{y(\gamma,\lambda)}\alpha_{j-1}$ ▷ Divide the lists into t strips of $\alpha_j m$ indices

  for $f_1(m)$ uniformly random permutation π of $\{1, \dots m\}$ **do**

    for $f_2(m)$ times **do**

      $r = (r_1, \dots, r_t) \leftarrow \left(\textsc{Random}(\mathbb{F}_2^{\alpha_j m})\right)_{j=1}^{t}$ s.t. $wt(r_j) = \alpha_j \frac{m}{2}$

      $\bar{L} \leftarrow \pi(L) + r$

      $\bar{R} \leftarrow \pi(R) + r$

      Remove from $\bar{L}$ and $\bar{R}$ all vectors that are not of weight $\alpha_j \frac{m}{2}$ on the j-th strip

      $C \leftarrow \textsc{RecursiveMO}(\bar{L}, \bar{R}, m, t, \epsilon, \gamma, \lambda, (\alpha)_{j=1}^{t}, 1)$

      if $C \neq \bot$ **then return** C

  return $\bot$

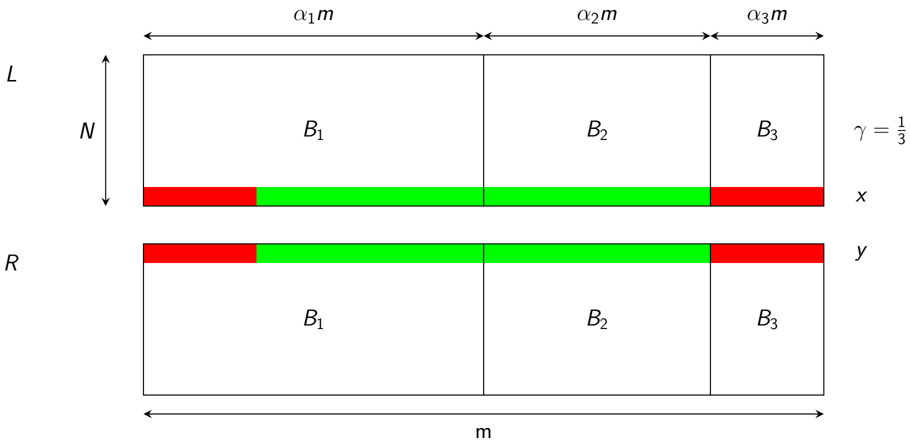
u_j : the $|B_j|$-bit vector of u's entries indexed by B_j

Double rerandomisation

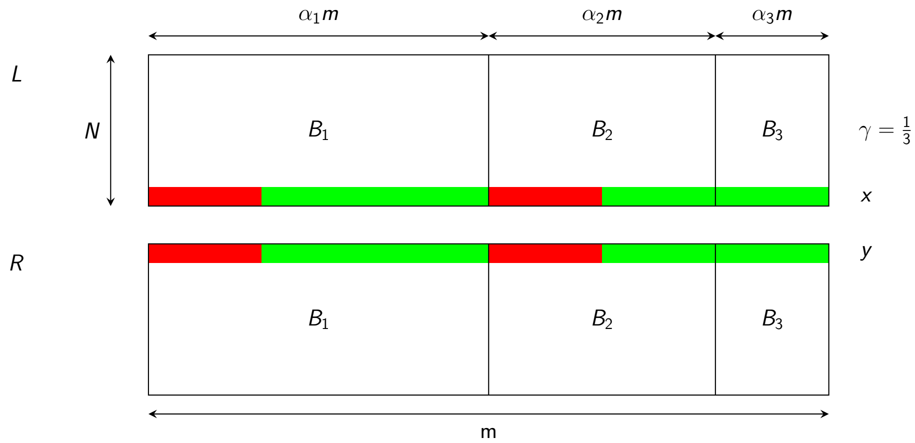
The recursive search requires the particular solution (x, y) to satisfy the following conditions. For all $1 \leq j \leq t$:

1. $\mathrm{d}(x_j, y_j) = \gamma \left| B_j \right| = \gamma \alpha_j m$;
2. $wt\left(x_j\right) = wt\left(y_j\right) = \frac{\alpha_j m}{2}$.

$t = 3$

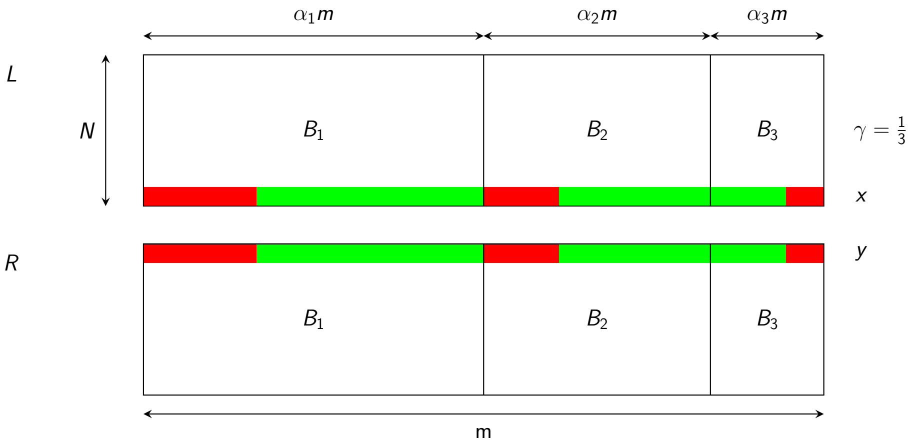

$\alpha_1 m$
$\alpha_2 m$
$\alpha_3 m$
L
N
B_1
B_2
B_3
$\gamma = \frac{1}{3}$
x
y
R
B_1
B_2
B_3
m

$t = 3$



$\alpha_1 m$
$\alpha_2 m$
$\alpha_3 m$
L
N
B_1
B_2
B_3
$\gamma = \frac{1}{3}$
x
R
y
B_1
B_2
B_3
m

$t = 3$



$\alpha_1 m$
$\alpha_2 m$
$\alpha_3 m$
L
N
B_1
B_2
B_3
$\gamma = \frac{1}{3}$
x
y
R
B_1
B_2
B_3
m

Algorithm 1 The MO algorithm

function $\mathrm{MO}(\ L, R, \lambda, \gamma, m, t, \epsilon\ )$

  $y(\gamma, \lambda) \leftarrow (1-\gamma)\left(1 - H\left(\frac{H^{-1}(1-\lambda)-\gamma/2}{1-\gamma}\right)\right)$

  $\alpha_1 \leftarrow (y(\gamma,\lambda) - \lambda + \epsilon/2)/y(\gamma,\lambda)$

  for $2 \leq j \leq t$ **do**

    $\alpha_j \leftarrow \frac{\lambda}{y(\gamma,\lambda)}\alpha_{j-1}$ ▷ Divide the lists into t strips of $\alpha_j m$ indices

  for $f_1(m)$ uniformly random permutation π of $\{1, \ldots m\}$ **do**

    for $f_2(m)$ times **do**

      $r = (r_1, \ldots, r_t) \leftarrow \left(\mathrm{Random}(\mathbb{F}_2^{\alpha_j m})\right)_{j=1}^{t}$ s.t. $wt(r_j) = \alpha_j \frac{m}{2}$

      $\bar{L} \leftarrow \pi(L) + r$

      $\bar{R} \leftarrow \pi(R) + r$

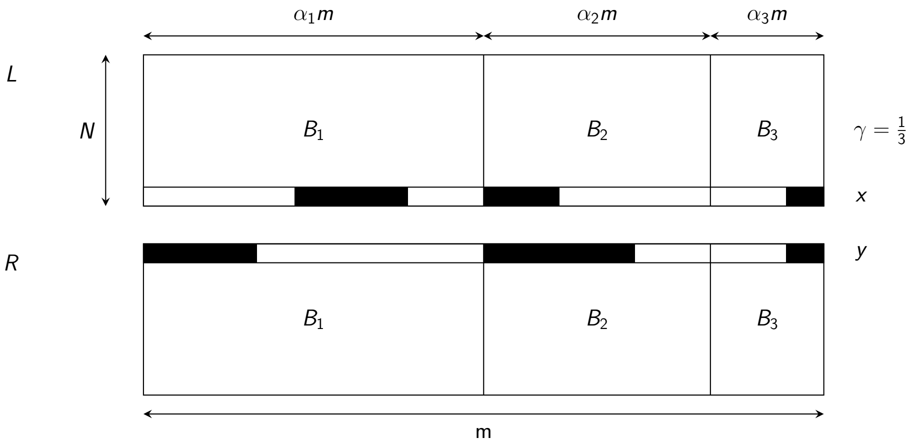
      Remove from $\bar{L}$ and $\bar{R}$ all vectors that are not of weight $\alpha_j \frac{m}{2}$ on the j-th strip

      $C \leftarrow \mathrm{RecursiveMO}(\bar{L}, \bar{R}, m, t, \epsilon, \gamma, \lambda, (\alpha)_{j=1}^{t}, 1)$

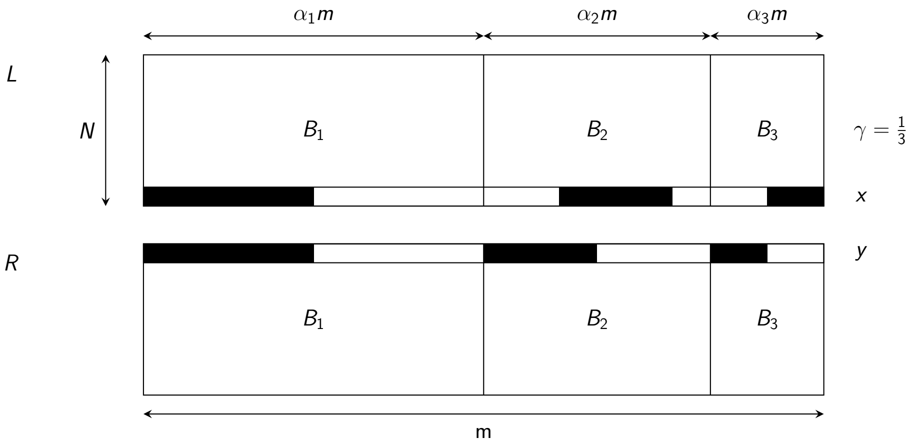
      if $C \neq \perp$ **then return** C

  return $\perp$

$t = 3$


$\alpha_1 m$
$\alpha_2 m$
$\alpha_3 m$
L
N
B_1
B_2
B_3
$\gamma = \frac{1}{3}$
x
R
y
B_1
B_2
B_3
m

$t = 3$



$\alpha_1 m$
$\alpha_2 m$
$\alpha_3 m$
L
N
B_1
B_2
B_3
$\gamma = \frac{1}{3}$
x
y
R
B_1
B_2
B_3
m

Algorithm 1 The MO algorithm

function $\mathrm{MO}(L, R, \lambda, \gamma, m, t, \epsilon)$

$y(\gamma,\lambda) \leftarrow (1-\gamma)\left(1 - H\left(\frac{H^{-1}(1-\lambda)-\gamma/2}{1-\gamma}\right)\right)$

$\alpha_1 \leftarrow (y(\gamma,\lambda) - \lambda + \epsilon/2)/y(\gamma,\lambda)$

for $2 \leq j \leq t$ **do**

$\alpha_j \leftarrow \frac{\lambda}{y(\gamma,\lambda)}\alpha_{j-1}$ ▷ Divide the lists into t strips of $\alpha_j m$ indices

for $f_1(m)$ uniformly random permutation π of $\{1,\dots m\}$ **do**

for $f_2(m)$ times **do**

$r = (r_1,\dots,r_t) \leftarrow \left(\textsc{Random}(\mathbb{F}_2^{\alpha_j m})\right)_{j=1}^{t}$ s.t. $wt(r_j) = \alpha_j \frac{m}{2}$

$\bar{L} \leftarrow \pi(L) + r$

$\bar{R} \leftarrow \pi(R) + r$

Remove from $\bar{L}$ and $\bar{R}$ all vectors that are not of weight $\alpha_j \frac{m}{2}$ on the j-th strip

$C \leftarrow \textsc{RecursiveMO}(\bar{L}, \bar{R}, m, t, \epsilon, \gamma, \lambda, (\alpha)_{j=1}^{t}, 1)$

if $C \neq \perp$ **then return** C

return $\perp$

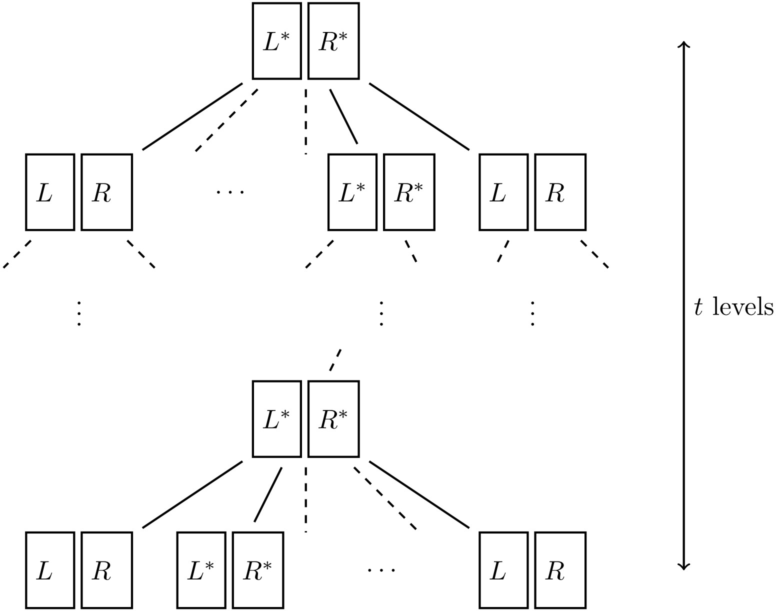
Lemma 1 [BDH25]

Let N_{it} the number of iterations of the double rerandomization. The particular solution (x, y) satisfies Conditions i) and ii) in at least one of the iterations with probability greater than $1/4$ only if

$$N_{it} = f_1(m)f_2(m) \geq \frac{1}{8\sqrt{2}} \left(\frac{\pi^{\frac{3}{2}}}{2}\right)^t m^{t-\frac{1}{2}} (\gamma(1-\gamma))^{t-\frac{1}{2}} \left(\frac{y(\gamma,\lambda) - \lambda + \frac{\epsilon}{2}}{y(\gamma,\lambda)}\right)^t \left(\frac{\lambda}{y(\gamma,\lambda)}\right)^{\frac{t(t-1)}{2}}.$$

Algorithm 1 The MO algorithm

function $\mathrm{MO}(L, R, \lambda, \gamma, m, t, \epsilon)$
    $y(\gamma,\lambda) \leftarrow (1-\gamma)\left(1 - H\left(\frac{H^{-1}(1-\lambda)-\gamma/2}{1-\gamma}\right)\right)$
    $\alpha_1 \leftarrow (y(\gamma,\lambda) - \lambda + \epsilon/2)/y(\gamma,\lambda)$
    for $2 \leq j \leq t$ **do**
        $\alpha_j \leftarrow \frac{\lambda}{y(\gamma,\lambda)}\alpha_{j-1}$ ▷ Divide the lists into t strips of $\alpha_j m$ indices
    for $f_1(m)$ uniformly random permutation π of $\{1, \ldots m\}$ **do**
        for $f_2(m)$ times **do**
            $r = (r_1, \ldots, r_t) \leftarrow \left(\textsc{Random}(\mathbb{F}_2^{\alpha_j m})\right)_{j=1}^{t}$ s.t. $wt(r_j) = \alpha_j \frac{m}{2}$
            $\bar{L} \leftarrow \pi(L) + r$
            $\bar{R} \leftarrow \pi(R) + r$
            Remove from $\bar{L}$ and $\bar{R}$ all vectors that are not of weight $\alpha_j \frac{m}{2}$ on the j-th strip
            $C \leftarrow \textsc{RecursiveMO}(\bar{L}, \bar{R}, m, t, \epsilon, \gamma, \lambda, (\alpha)_{j=1}^{t}, 1)$
            if $C \neq \perp$ **then return** C
    return $\perp$



L^*
R^*
L
R
$\cdots$
L^*
R^*
L
R
L^*
R^*
L
R
L^*
R^*
$\cdots$
L
R
t levels

At a node at depth j, filtering is performed on B_j

Filtering on B_j

- Pick a random subset A of $\frac{\alpha_j m}{2}$ indices inside strip j
- $L' \leftarrow \{u \in L \text{ s.t. } wt(u_A) = H^{-1}(1-\lambda)\frac{\alpha_j m}{2}\}$
- $R' \leftarrow \{v \in R \text{ s.t. } wt(v_A) = H^{-1}(1-\lambda)\frac{\alpha_j m}{2}\}$

$$\mathbb{P}\left[(x,y) \in L' \times R' \mid (x,y) \in L \times R)\right] = \frac{1}{s_j} = \tilde{\mathcal{O}}\left(2^{-\alpha_j y(\gamma,\lambda)m}\right)$$

$$\mathbb{P}\left[u \in L' \mid u \in L\right] = p_j = \tilde{\mathcal{O}}\left(2^{-\lambda\alpha_j m}\right)$$

$$\mathbb{P}\left[(x,y) \in L' \times R' \mid (x,y) \in L \times R)\right] = \frac{1}{s_j} = \tilde{\mathcal{O}}\left(2^{-\alpha_j y(\gamma,\lambda) m}\right)$$

$$\mathbb{P}\left[u \in L' \mid u \in L\right] = p_j = \tilde{\mathcal{O}}\left(2^{-\lambda \alpha_j m}\right)$$

- Filtering ms_j times on strip B_j
  $\implies$ the particular solution (x,y) survives with overwhelming probability.

$$\mathbb{P}\left[(x,y) \in L' \times R' \mid (x,y) \in L \times R)\right] = \frac{1}{s_j} = \tilde{\mathcal{O}}\left(2^{-\alpha_j y(\gamma,\lambda) m}\right)$$

$$\mathbb{P}\left[u \in L' \mid u \in L\right] = p_j = \tilde{\mathcal{O}}\left(2^{-\lambda \alpha_j m}\right)$$

- Filtering ms_j times on strip B_j
  $\implies$ the particular solution (x, y) survives with overwhelming probability.
- $\mathbb{E}[\#R'] = \mathbb{E}[\#L'] = N \prod_{i=1}^{j} p_i = \tilde{\mathcal{O}}\left(2^{\lambda m \left(1 - \sum_{i=1}^{j} \alpha_j\right)}\right)$

$$\mathbb{P}\left[(x,y) \in L' \times R' \mid (x,y) \in L \times R)\right] = \frac{1}{s_j} = \tilde{\mathcal{O}}\left(2^{-\alpha_j y(\gamma,\lambda) m}\right)$$

$$\mathbb{P}\left[u \in L' \mid u \in L\right] = p_j = \tilde{\mathcal{O}}\left(2^{-\lambda \alpha_j m}\right)$$

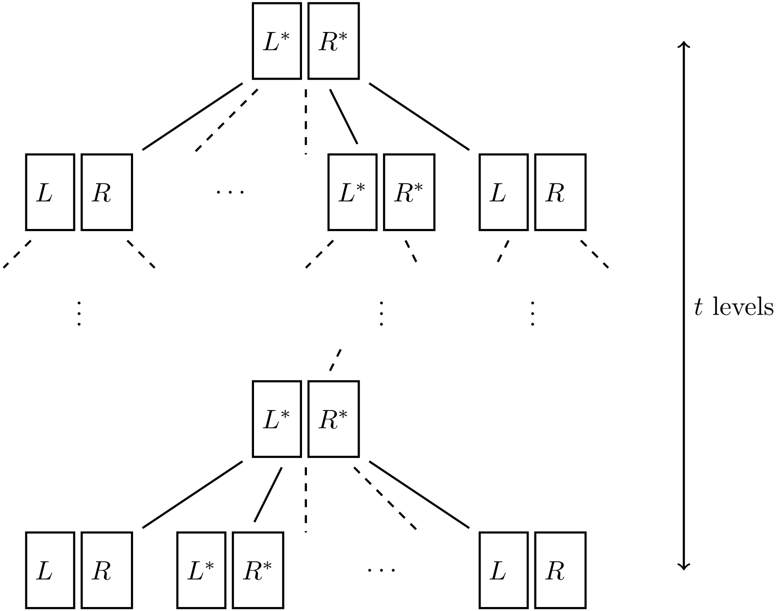
- Filtering ms_j times on strip B_j
  $\implies$ the particular solution (x,y) survives with overwhelming probability.
- $\mathbb{E}[\#R'] = \mathbb{E}[\#L'] = N \prod_{i=1}^{j} p_i = \tilde{\mathcal{O}}\left(2^{\lambda m \left(1 - \sum_{i=1}^{j} \alpha_j\right)}\right)$
- Let $X = \#R'$ (Tchebitchev inequality)
  $\implies \mathbb{P}\left[X - \mathbb{E}[X] \geq 2^{\frac{\epsilon}{2} m} \mathbb{E}[X]\right] \leq 2^{-\epsilon m}$

Algorithm 2 RecursiveMO

function RecursiveMO$(L, R, m, t, \epsilon, \lambda, \gamma, (\alpha)_1^t, j)$
  if $j = t + 1$ **then**
    return QuadraticNN$(L, R, \gamma m)$
  $C \leftarrow \perp$

$$s_j \leftarrow \frac{\binom{\alpha_j m}{\frac{1}{2}\alpha_j m}}{\left(\binom{(1-\gamma)\frac{\alpha_j m}{2}}{(1-h-\frac{\gamma}{2})\frac{\alpha_j m}{2}}\binom{(1-\gamma)\frac{\alpha_j m}{2}}{(h-\frac{\gamma}{2})\frac{\alpha_j m}{2}}\binom{\gamma\frac{\alpha_j m}{2}}{\frac{\gamma}{2}\frac{\alpha_j m}{2}}\right)^2}$$

  for ms_j times **do**
    Pick a random subset A of $\frac{\alpha_j m}{2}$ indices inside strip j
    $L' \leftarrow \{u \in L \text{ s.t. } wt(u_A) = H^{-1}(1-\lambda)\frac{\alpha_j m}{2}\}$
    $R' \leftarrow \{v \in R \text{ s.t. } wt(v_A) = H^{-1}(1-\lambda)\frac{\alpha_j m}{2}\}$
    if $|L'|$ and $|R'|$ are not too big **then**
      $x \leftarrow$ RecursiveMO$(L', R', m, t, \epsilon, \lambda, \gamma, (\alpha)_1^t, j+1)$
      if $x \neq \perp$ **then**
        return x
  return C



L^* R^*
L R
L^* R^*
L R
L^* R^*
L R
L^* R^*
L R
t levels

For all $2 \leq j \leq t$

Total complexity at depth j

$$T_j = \tilde{O}\left((2^m)^{\sum_{i=1}^{j-1} \alpha_i y(\gamma,\lambda) + \alpha_j y(\gamma,\lambda) + \lambda \left(1 - \sum_{i=1}^{j-1} \alpha_i\right) + \frac{\epsilon}{2}} \right)$$

For all $2 \le j \le t$

Total complexity at depth j

$$T_j = \tilde{\mathcal{O}}\left((2^m)^{\sum_{i=1}^{j-1} \alpha_i y(\gamma,\lambda) + \alpha_j y(\gamma,\lambda) + \lambda\left(1 - \sum_{i=1}^{j-1} \alpha_i\right) + \frac{\epsilon}{2}} \right)$$

For all $2 \leq j \leq t$

Total complexity at depth j

$$T_j = \tilde{\mathcal{O}}\left((2^m)^{\sum_{i=1}^{j-1} \alpha_i y(\gamma,\lambda) + \alpha_j y(\gamma,\lambda) + \lambda\left(1 - \sum_{i=1}^{j-1} \alpha_i\right) + \frac{\epsilon}{2}} \right)$$

Case $j = 1$: $T_1 = \tilde{\mathcal{O}}(2^{\lambda m + \alpha_1 y(\gamma,\lambda) m + \frac{\epsilon}{2} m})$

Depth $t+1 \implies$ Exhaustive search

Maximum list size: $\tilde{\mathcal{O}}\left((2^m)^{\lambda(1-\sum_{i=1}^{t}\alpha_i)+\frac{\epsilon}{2}}\right) = \tilde{\mathcal{O}}\left(2^{\frac{\epsilon}{2}m}\right)$

Number of node: $\tilde{\mathcal{O}}\left((2^m)^{\sum_{i=1}^{t}\alpha_i y(\gamma,\lambda)}\right) = \tilde{\mathcal{O}}\left(2^{y(\gamma,\lambda)m}\right)$

Total complexity at depth $t+1$

$$T_{t+1} = \tilde{\mathcal{O}}\left(2^{(y(\gamma,\lambda)+\epsilon)m}\right)$$

Algorithm 3 RecursiveMO

function RecursiveMO$(L, R, m, t, \epsilon, \lambda, \gamma, (\alpha)_1^t, j)$

  if $j = t + 1$ **then**

    return QuadraticNN$(L, R, \gamma m)$

  $C \leftarrow \perp$

  $$s_j \leftarrow \frac{\binom{\alpha_j m}{\frac{1}{2}\alpha_j m}}{\left(\binom{(1-\gamma)\frac{\alpha_j m}{2}}{(1-h-\frac{\gamma}{2})\frac{\alpha_j m}{2}}\binom{(1-\gamma)\frac{\alpha_j m}{2}}{(h-\frac{\gamma}{2})\frac{\alpha_j m}{2}}\binom{\gamma\frac{\alpha_j m}{2}}{\frac{\gamma}{2}\frac{\alpha_j m}{2}}\right)^2}$$

  for ms_j times **do**

    Pick a random subset A of $\frac{\alpha_j m}{2}$ indices inside strip j

    $L' \leftarrow \{u \in L \text{ s.t. } wt(u_A) = H^{-1}(1-\lambda)\frac{\alpha_j m}{2}\}$

    $R' \leftarrow \{v \in R \text{ s.t. } wt(v_A) = H^{-1}(1-\lambda)\frac{\alpha_j m}{2}\}$

    if $|L'|$ and $|R'|$ are not too big **then**

      $x \leftarrow$ RecursiveMO$(L', R', m, t, \epsilon, \lambda, \gamma, (\alpha)_1^t, j+1)$

      if $x \neq \perp$ **then**

        return x

  return C

To conclude, choosing:

1. $\alpha_{j+1} = \frac{\lambda \alpha_j}{y(\gamma,\lambda)}$ for all $j \in \{1, \ldots, t-1\}$
2. $t = \left\lceil \frac{\log(2(y(\gamma,\lambda)-\lambda)/\epsilon+1)}{\log(y(\gamma,\lambda)/\lambda)} \right\rceil$

leads to

$$
\begin{aligned}
T_1 &= \tilde{\mathcal{O}}\left(2^{(y(\gamma,\lambda)+\epsilon)m}\right)\\
T_2 &= \tilde{\mathcal{O}}\left(2^{(y(\gamma,\lambda)+\epsilon)m}\right)\\
\ldots &= \tilde{\mathcal{O}}\left(2^{(y(\gamma,\lambda)+\epsilon)m}\right)\\
T_t &= \tilde{\mathcal{O}}\left(2^{(y(\gamma,\lambda)+\epsilon)m}\right)\\
T_{t+1} &= \tilde{\mathcal{O}}\left(2^{(y(\gamma,\lambda)+\epsilon)m}\right)
\end{aligned}
$$

- The model : $\mathcal{M}_{Alea}\left(\mathbb{F}_2, 2^{\lambda m}\right)$
- The expected number of solutions : $E = \Theta\left(m^{-\frac{1}{2}} 2^{m(2\lambda + \mathsf{H}(\gamma) - 1)}\right)$
- Fonctions LSH
- A lower bound in a nearby model [KL21] : $2^{\frac{\lambda}{1-\gamma} m}$

Thank you