Private Information Retrieval and Batch Codes

Bimal Roy
Indian Statistical Institute, Kolkata.

Outline of the talk

1. Private Information Retrieval
2. Batch Codes
   - Problem
   - Definition and Examples
3. Combinatorial Batch Codes
   - Setting
   - Problem 1
   - Results
   - Problem 2
   - Results

Outline of the talk

Problem Description (informal)

- A database is stored in a set of servers.
- Goal is to retrieve a record from the database without revealing the identity of the retrieved record to the servers.

Protocol Description (informal)

- User generates a set of **queries** from the **index** and a **random string**.
- User sends the queries to the servers.
- Each server sends back an **answer**.

Correctness requirement: User should be able to get the queried record from the index, the random string, and the answers.

Privacy requirement: Server gets no information about the **queried index**.

Protocol Description (formal)

Setting:

- One-round protocol involving a user and k servers.
- Each server contains a database $x \in \{0,1\}^n$ (identical for each server).
- User wants to retrieve the i-th entry x_i of the database x.
- Picks a random string r and generates k queries $q_j = Q_j(i, r), j \in [k]$, and sends q_j to j-th server.
- For each $j \in [k]$, the j-th sever computes answer $a_j = A_j(q_j, x)$ and sends a_j to the user.
- User computes $x_i = R(a_1, \ldots, a_k, i, r)$.

The functions $Q_j, A_j, j \in [k]$ and R are described in the following definition.

Private Information Retrieval(PIR):

A k-server PIR scheme for database length n consists of

- k query functions $Q_j(.,.), j \in [k]$;
- k answer functions $A_j(.,.), j \in [k]$;
- a reconstruction function $R(.,\ldots,.)$, R has $k+2$ arguments,

Correctness: For every $x \in \{0,1\}^n$, every $i \in [n]$, and for every r

$$R(a_1, \ldots, a_k, i, r) = x_i.$$

Privacy: For every $i_1, i_2 \in [n]$ and for every $j \in [k]$,

$$Pr\{Q_j(i_1, r) = q\} = Pr\{Q_j(i_2, r) = q\}.$$

Solution:

1. Trivial solution: Send the entire database; communication overhead $O(n)$.
2. A 2-server, $O(\sqrt{n})$ solution:
   - Store the database (a string of length n) as $\sqrt{n} \times \sqrt{n}$ table in 2 servers S_1 and S_2.
   - User wants to retrieve the entry $(i, j), 1 \leq i \leq \sqrt{n}, 1 \leq j \leq \sqrt{n}$.
   - User randomly selects a subset of S of $\{1, \ldots, \sqrt{n}\}$.
   - User sends S (as a bit string of length $\sqrt{n}$) to S_1.
   - If $j \in S$, user sends $S \setminus \{j\}$ to S_2 (as a bit string of length $\sqrt{n}$), otherwise sends $S \bigcup \{j\}$ to S_2.
   - For each row the servers compute xor of the indices (column entries) present in the set sent by the user, and send the row-wise sum as a $\sqrt{n}$ bit string.
   - User computes xor of the i-th entries of the bit strings sent by the servers.

Example 1:

- Let the database be

a	b	c
d	e	f
g	h	i

$a \dots i \in F_2$.

and the servers be S_1 and S_2.

- User wants to retrieve the entry $(3, 1)$.
- Randomly selects a subset of $\{1, 2, 3\}$. Let it be $\{1, 3\}$.
- User sends $\{1, 3\}$ to S_1.
- User sends $\{3\}$ to S_2.
- S_1 sends $a \oplus c$, $d \oplus f$, $g \oplus i$ to user; S_2 sends c, f, i to user.
- User computes $g \oplus i \oplus i = g$.

Example 2:

- Let the database be

a	b	c
d	e	f
g	h	i

$a \dots i \in F_2$.

and the servers be S_1 and S_2.

- User wants to retrieve the entry $(3, 1)$.
- Randomly selects a subset of $\{1, 2, 3\}$. Let it be $\{2, 3\}$.
- User sends $\{2, 3\}$ to S_1.
- User sends $\{1, 2, 3\}$ to S_2.
- S_1 sends $b \oplus c$, $e \oplus f$, $h \oplus i$ to user; S_2 sends $a \oplus b \oplus c$, $d \oplus e \oplus f$, $g \oplus h \oplus i$ to user.
- User computes $h \oplus i \oplus g \oplus h \oplus i = g$.

Implementation bottleneck

Implementation bottleneck

Computational overhead: Computation must be $O(n)$ for a database of size n - huge for large databases.

Implementation bottleneck

Computational overhead: Computation must be $O(n)$ for a database of size n - huge for large databases.

Workaround

1. Preprocessing.

Implementation bottleneck

Computational overhead: Computation must be $O(n)$ for a database of size n - huge for large databases.

Workaround

1. Preprocessing.
2. Amortizing - amortize the computational overhead over k queries.

Implementation bottleneck

Computational overhead: Computation must be $O(n)$ for a database of size n - huge for large databases.

Workaround

1. Preprocessing.
2. Amortizing - amortize the computational overhead over k queries.

   Can the computational cost be made $< O(kn)$ for k queries?

Implementation bottleneck

Computational overhead: Computation must be $O(n)$ for a database of size n - huge for large databases.

Workaround

1. Preprocessing.
2. Amortizing - amortize the computational overhead over k queries.

   Can the computational cost be made $< O(kn)$ for k queries?

Batch codes

Outline of the talk

Outline of the talk

Problem

(n, N, k, m, t)-CBC:

Problem

(n, N, k, m, t)-CBC:

- n different data items

Problem

(n, N, k, m, t)-CBC:

- n different data items
- m servers

Problem

(n, N, k, m, t)-CBC:

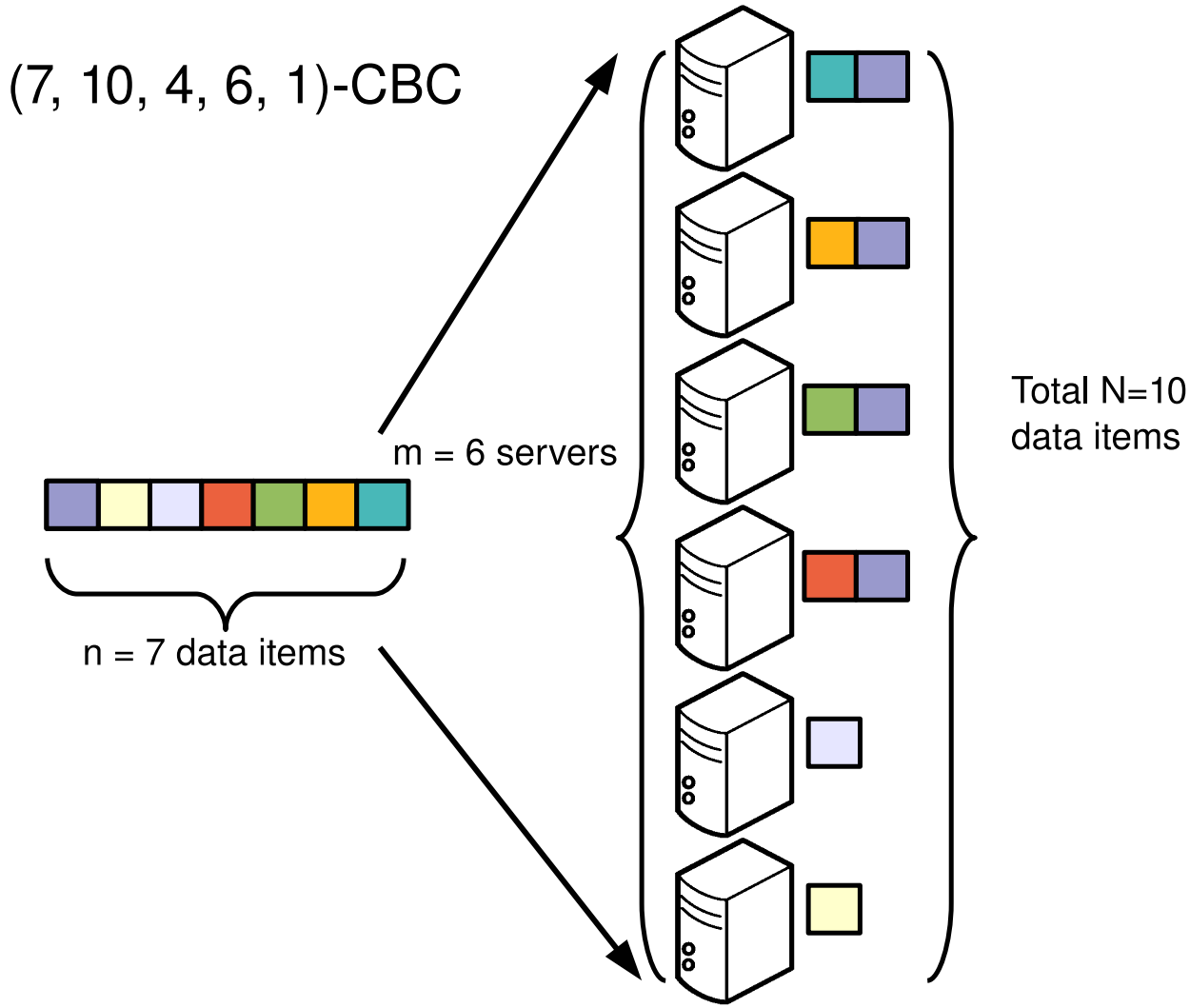
- n different data items
- m servers
- Any $k(\leq n)$ data items are to be retrieved by reading at most $t(\leq k)$ items from each server

Problem

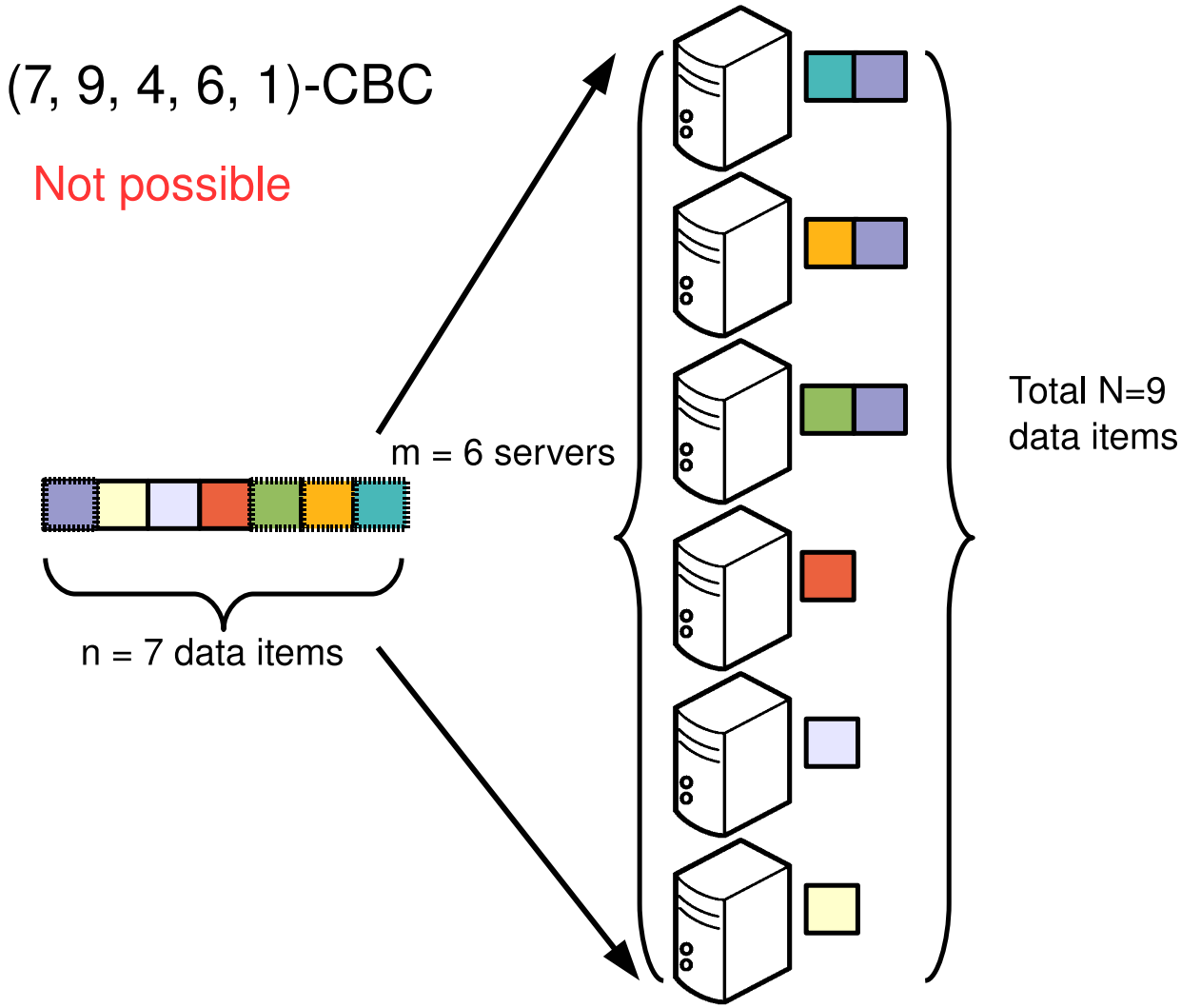
(n, N, k, m, t)-CBC:

- n different data items
- m servers
- Any $k(\leq n)$ data items are to be retrieved by reading at most $t(\leq k)$ items from each server
- Total storage N

An Example



Examples(contd..)



Outline of the talk

Batch Code:Definition

Definition

[Ishai et al., 2004] An (n, N, k, m, t) batch code over an alphabet Σ is defined by an encoding function $C : \Sigma^n \to (\Sigma^*)^m$ (each output of which is called a bucket) and a decoding algorithm A such that

1. The total length of all m buckets is N (where the length of each bucket is independent of x).
2. For any $x \in \Sigma^n$ and $\{i_1, \ldots, i_k\} \subseteq [n], A(C(x), i_1, \ldots, i_k) = (x_{i_1}, \ldots, x_{i_k})$, and A probes at most t symbols from each bucket in $C(x)$ (whose positions are determined by $i_1, \ldots, i_k$).

Ishai, Y., Kushilevitz, E., Ostrovsky, R., and Sahai, A. (2004).
Batch codes and their applications.
In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, volume 36, pages 262–271.

Combinatorial Batch Codes (CBC)

Combinatorial Batch Codes (CBC)

- Encoding: Assignment of items to servers.

Combinatorial Batch Codes (CBC)

- Encoding: Assignment of items to servers.
- Decoding: Retrieving items from servers.

Combinatorial Batch Codes (CBC)

- Encoding: Assignment of items to servers.
- Decoding: Retrieving items from servers.

- Advantage: Time required for encoding / decoding is less.

Combinatorial Batch Codes (CBC)

- Encoding: Assignment of items to servers.
- Decoding: Retrieving items from servers.

- Advantage: Time required for encoding / decoding is less.
- Disadvantage: May require more space.

CBC vs. NonCBC

Example([Ishai et al., 2004]):

- **Problem:** Construct a batch code for a binary string of length $n(14)$, with the following parameters
  $m = 3, k = 2, t = 1, N = 1.5n(21)$.

CBC vs. NonCBC

Example([Ishai et al., 2004]):

- **Problem:** Construct a batch code for a binary string of length $n(14)$, with the following parameters
  $m = 3, k = 2, t = 1, N = 1.5n(21)$.
- CBC not possible-

CBC vs. NonCBC

Example([Ishai et al., 2004]):

- **Problem:** Construct a batch code for a binary string of length $n(14)$, with the following parameters $m = 3, k = 2, t = 1, N = 1.5n(21)$.
- CBC not possible-
  - At least $\frac{n}{2}(7)$ items have single instances.

CBC vs. NonCBC

Example([Ishai et al., 2004]):

- **Problem:** Construct a batch code for a binary string of length $n(14)$, with the following parameters $m = 3, k = 2, t = 1, N = 1.5n(21)$.
- CBC not possible-
  - At least $\frac{n}{2}(7)$ items have single instances.
  - At least one server will have $\frac{1}{3}$ rd (3) of these items.

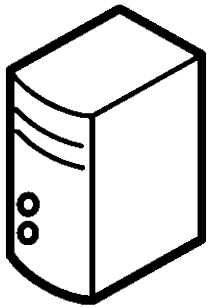
CBC vs. NonCBC

Example([Ishai et al., 2004]):

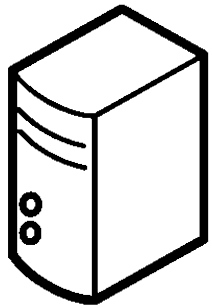
- **Problem:** Construct a batch code for a binary string of length $n(14)$, with the following parameters
  $m = 3, k = 2, t = 1, N = 1.5n(21)$.
- CBC not possible-
  - At least $\frac{n}{2}(7)$ items have single instances.
  - At least one server will have $\frac{1}{3}$ rd (3) of these items.
  - Can not retrieve any 2 of the above items.

CBC vs. Non-CBC(contd..)

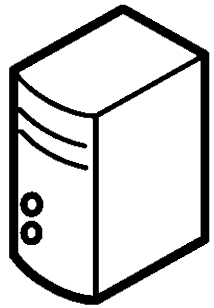
A Non-CBC solution -



X1
X2
X3
X4
X5
X6
X7



X8
X9
X10
X11
X12
X13
X14



X1 ⊕ X8
X2 ⊕ X9
X3 ⊕ X10
X4 ⊕ X11
X5 ⊕ X12
X6 ⊕ X13
X7 ⊕ X14

Amortization in PIR:

Computation required to retrieve 2 items from 3 servers when the database is simply replicated - $O(2n)$.
Batch code based scheme:

1. Encode the database according to the previous scheme.
2. Suppose the user wants to retrieve x_2 and x_4.
3. Run 1-server PIR on S_1, S_2, S_3 to retrieve x_2, x_{11} and $x_4 \oplus x_{11}$ respectively.

Total computation required is $3 \times O(.5n) = 1.5n$.

Outline of the talk

Outline of the talk

Combinatorial Batch Codes: Setting

Set system - $(\mathcal{S}, \mathcal{I})$

- Ground Set $(\mathcal{S})$ - Set of servers.
- Collection of subsets $(\mathcal{I})$ - Corresponds to items
  A subset corresponding to an item contains only those servers which store that item (may have multiple copies of same subset).

Paterson, M. B., Stinson, D. R., and Wei, R. (2009).
Combinatorial batch codes.
Advances in Mathematics of Communications, 3:13–27.

Setting (contd..)

Example:

- Servers $A, B, C; \Rightarrow \mathcal{S} = \{A, B, C\}$.
- A contains items a, b, c. B contains items a, b. C contains item b.
- $\mathcal{I} = \{\{A, B\}, \{A, B, C\}, \{A\}\}$.
  $\{A, B\} \Rightarrow a$.
  $\{A, B, C\} \Rightarrow b$.
  $\{A\} \Rightarrow c$.

Setting (contd..)

Notation and Terminology:

- (n, N, k, m)-CBC $\Rightarrow$ (n, N, k, m, t)-CBC with $t = 1$.
- $N(n, k, m) \Rightarrow$ Minimum value of N such that there is an $N(n, N, k, m)$-CBC.
- c-uniform CBC $\Rightarrow$ Each data item is stored in c servers $\Rightarrow$ Each subset of the set system consists of c elements.
- $n(m, c, k) \Rightarrow$ Maximum value of n for which there exists c-uniform (n, cn, m, k) - CBC.

Setting(contd..)

Definition

[Paterson et al., 2009] An (n, N, k, m) batch code is a set system $(\mathcal{S}, \mathcal{I})$, such that -

- $|\mathcal{S}| = m$.
- $|\mathcal{I}| = n$.
- Any subcollection of k subsets of $\mathcal{S}$ has a system of distinct representatives.
  or(*Hall's theorem*)
  Any subcollection of i subsets contains i elements for $1 \leq i \leq k$.

Outline of the talk

Problem

Given

Problem

Given

- Number of data items - n.

Problem

Given

- Number of data items - n.
- Number of servers - m.

Problem

Given

- Number of data items - n.
- Number of servers - m.
- Retrievability parameter - k.

Problem

Given

- Number of data items - n.
- Number of servers - m.
- Retrievability parameter - k.

Minimize the total storage N, i. e., Determine $N(n, k, m)$.

Outline of the talk

Explicit constructions and bounds for $N(n, k, m)$

Theorem ([Paterson et al., 2009])

$N(n, k, k) = kn - k(k-1)$

- k copies of each of any $n - k$ items in k servers.
- single copy of each of the remaining k items in k different servers.

Item \ Server	1	2	...	k
1	1	1	...	1
2	1	1	...	1
⋮	⋮	⋮	⋮	⋮
$n-k$	1	1	...	1
$n-k+1$	1	0	...	0
$n-k+2$	0	1	...	0
⋮	⋮	⋮	⋮	⋮
n	0	0	...	1

Constructions and bounds (contd..)

Theorem ([Paterson et al., 2009])

$N(m+1, k, m) = m + k$

- Single copy of each of any m items in m distinct servers
- k copies of the remaining item in any k servers.

Item \ Server	1	2	...	k	$k+1$	...	m
1	1	0	...	0	0	...	0
2	0	1	...	0	0	...	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
m	0	0	...	0	0	...	1
$m+1$	1	1	...	1	0	...	0

Constructions and bounds (contd..)

Theorem ([Paterson et al., 2009])

For $n \geq (k-1)\binom{m}{k-1}$, $N(n,k,m) = kn - (k-1)\binom{m}{k-1}$

Construction:

- Make $(k-1)$ copies of each element of $\binom{[m]}{k-1}$.
- Select arbitrary k-subsets of $[m]$ for the rest $(n-(k-1)\binom{m}{k-1})$ elements.

Constructions and bounds (contd..)

Theorem ([Brualdi et al., 2010], [Bujtás and Tuza, 2011])

Let k and m be integers with $2 \leq k \leq m$. Then

$$N(m+2,k,m) = \begin{cases} m+k-2+\lceil 2\sqrt{k+1} \rceil & \text{if} \quad m+1-k \geq \lceil \sqrt{k+1} \rceil, \\ 2m-2+\lceil 1+\frac{k+1}{m+1-k} \rceil & \text{if} \quad m+1-k < \lceil \sqrt{k+1} \rceil. \end{cases}$$

Brualdi, R. A., Kiernan, K. P., Meyer, S. A., and Schroeder, M. W. (2010).
Combinatorial batch codes and transversal matroids.
Advances in Mathematics of Communications, 4:419–431.

Bujtás, C. and Tuza, Z. (2011).
Optimal combinatorial batch codes derived from dual systems.
Miskolc Math. Notes, 12:11–23.

A lower bound

Define $U_{m,k,c} := \frac{(k-1)\binom{m}{c}}{\binom{k-1}{c}}$.

Theorem ([Bhattacharya et al., 2012])

Let $1 \leq c < c' \leq k-1$, be such that $n \leq U_{m,k,c}$ (hence $n \leq U_{m,k,c'}$), then $nc' - \lfloor \frac{(k-c)d'}{m-k+1} \rfloor \leq nc - \lfloor \frac{(k-c)d}{m-k+1} \rfloor$, where $d = U_{m,k,c} - n$ and $d' = U_{m,k,c'} - n$. In particular, if c is least integer such that $n \leq U_{m,k,c}$, then $nc - \lfloor \frac{(k-c)d}{m-k+1} \rfloor$ is the lower bound for $N(n,k,m)$.

Bhattacharya, S., Ruj, S., and Roy, B. (2012).
Combinatorial batch codes: A lower bound and optimal constructions.
Advances in Mathematics of Communications, 6:165–174.

Optimal construction 1

Optimal construction for the range $\binom{m}{k-2} \leq n \leq (k-1)\binom{m}{k-1}$.

Theorem ([Bhattacharya et al., 2012], [Bujtás and Tuza, 2011])

For $\binom{m}{k-2} \leq n \leq (k-1)\binom{m}{k-1}$, $N(n,k,m) = n(k-1) - \lfloor \frac{d}{m-k+1} \rfloor$. *Where,* $d = (k-1)\binom{m}{k-1} - n$.

Example.Let $m = 10, k = 7, n = 500, \Rightarrow N(500, 7, 10) = 2810$.


Bhattacharya, S., Ruj, S., and Roy, B. (2012).
Combinatorial batch codes: A lower bound and optimal constructions.
Advances in Mathematics of Communications, 6:165–174.


Bujtás, C. and Tuza, Z. (2011).
Optimal batch codes: Many items or low retrieval requirement.
Advances in Mathematics of Communications, 5:529–541.

Optimal construction 2

Optimal and near optimal construction for the range $\binom{m}{k-2} - (m-k+1)A(m,4,k-3) \leq n \leq \binom{m}{k-2}$ for $k \geq 5$, where $A(m,4,k-3)$ is the maximum number of codewords of a binary constant weight code of length m, weight $k-3$ and Hamming distance 4.

Theorem ([Bhattacharya et al., 2012])

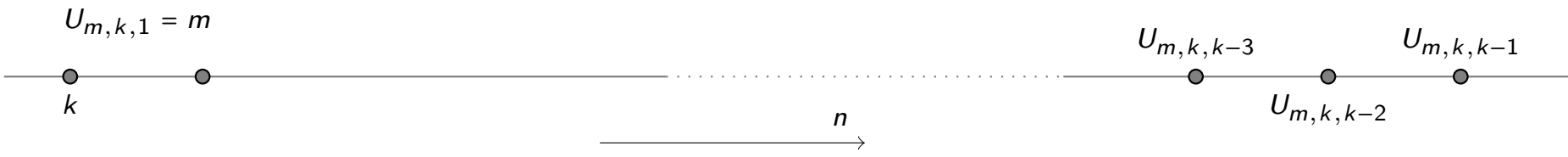
Let $\binom{m}{k-2} - (m-k+1)A(m,4,k-3) \leq n \leq \binom{m}{k-2}$. *Then*

$N(n,k,m) = n(k-2) - \left\lfloor \frac{2(\binom{m}{k-2}-n)}{m-k+1} \right\rfloor$ *for* $0 \leq (\binom{m}{k-2} - n) \mod (m-k+1) < \frac{m-k+1}{2}$ *and* $N(n,k,m) \leq n(k-2) - 2\left\lfloor \frac{\binom{m}{k-2}-n}{m-k+1} \right\rfloor$ *for* $\frac{m-k+1}{2} \leq (\binom{m}{k-2} - n) \mod (m-k+1) < m-k+1$.

Example. Let $m = 10, k = 7, n = 200, \Rightarrow A(10,4,4) = 30, N(200,7,10) = 974$.

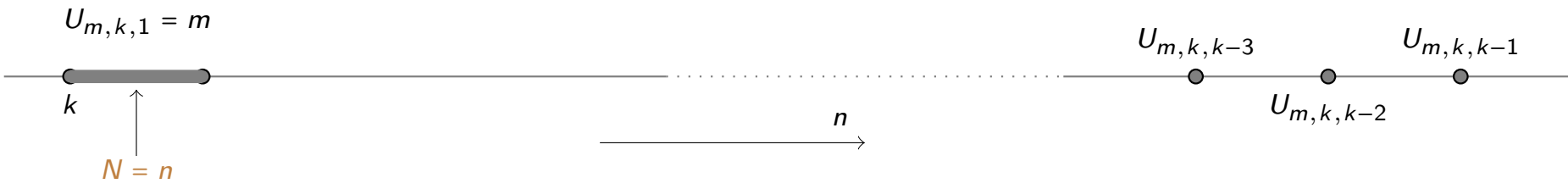
Present status

Optimal construction:



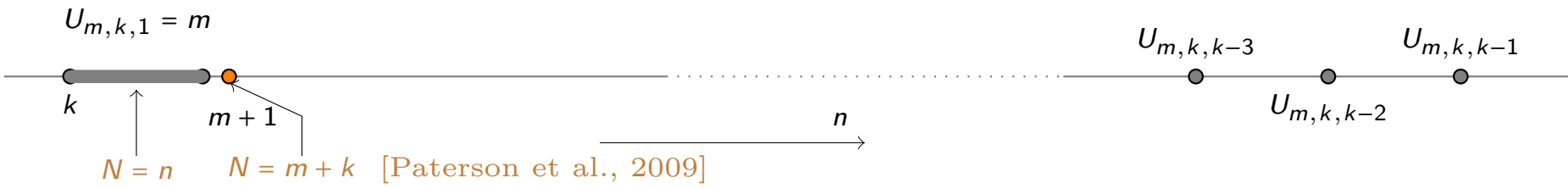
Present status

Optimal construction:



Present status

Optimal construction:

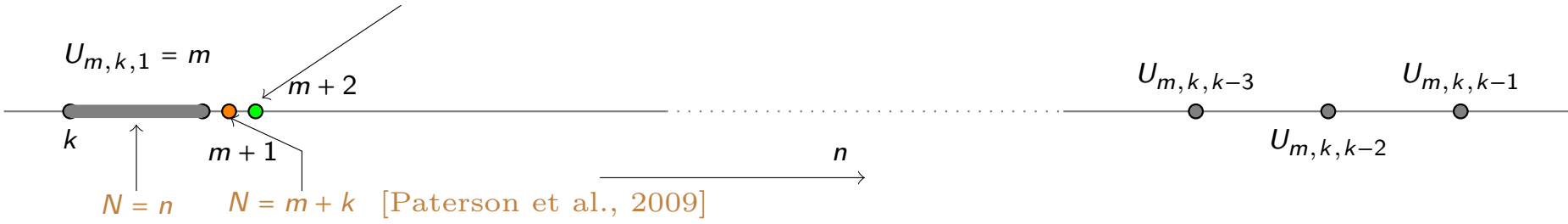


[Paterson et al., 2009]

Present status

Optimal construction:

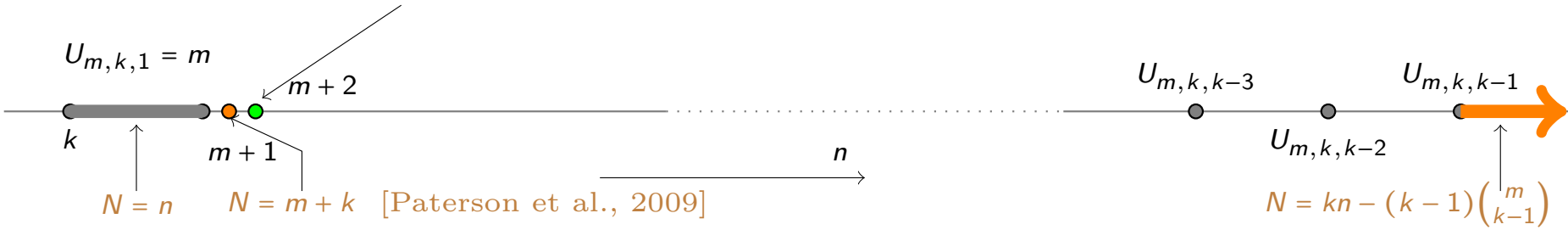
$N = m + k - 2 + \lceil 2\sqrt{k+1} \rceil$ [Brualdi et al., 2010], [Bujtás and Tuza, 2011]



Present status

Optimal construction:

$N = m + k - 2 + \lceil 2\sqrt{k+1} \rceil$ [Brualdi et al., 2010], [Bujtás and Tuza, 2011]



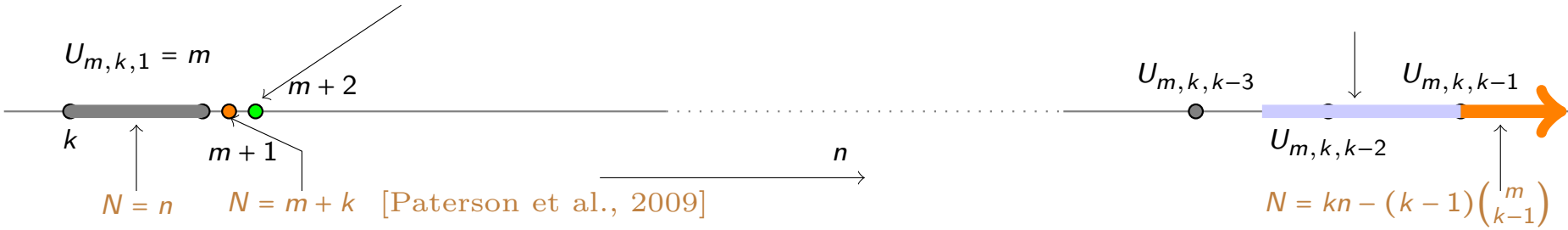
Present status

Optimal construction:

$N = nc - \lfloor \frac{(k-c)d}{m-k+1} \rfloor$

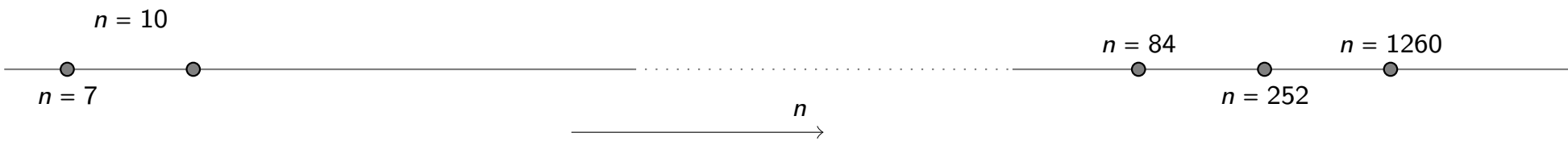
$N = m + k - 2 + \lceil 2\sqrt{k+1} \rceil$ [Brualdi et al., 2010], [Bujtás and Tuza, 2011]

[Bhattacharya et al., 2012]



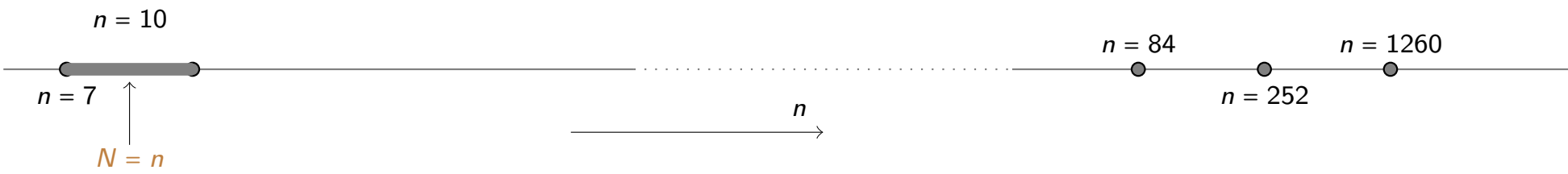
Present status

Optimal construction for $m = 10, k = 7$:



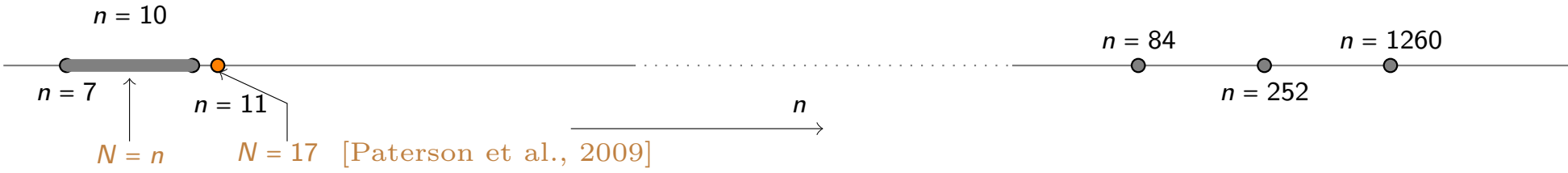
Present status

Optimal construction for $m = 10, k = 7$:



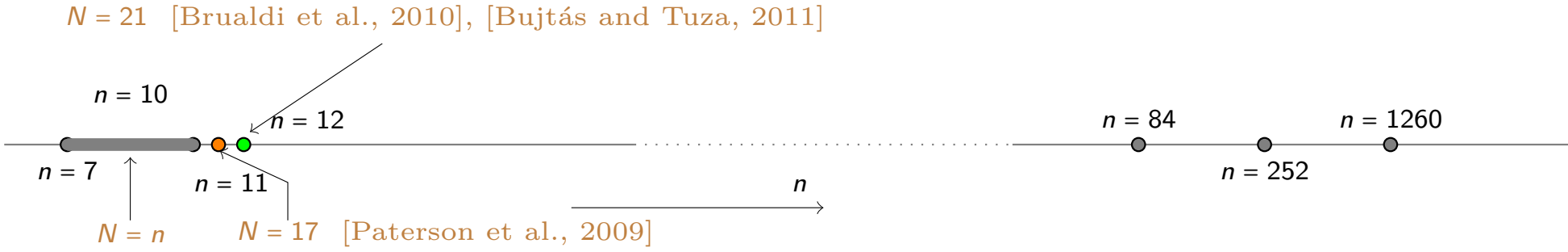
Present status

Optimal construction for $m = 10, k = 7$:



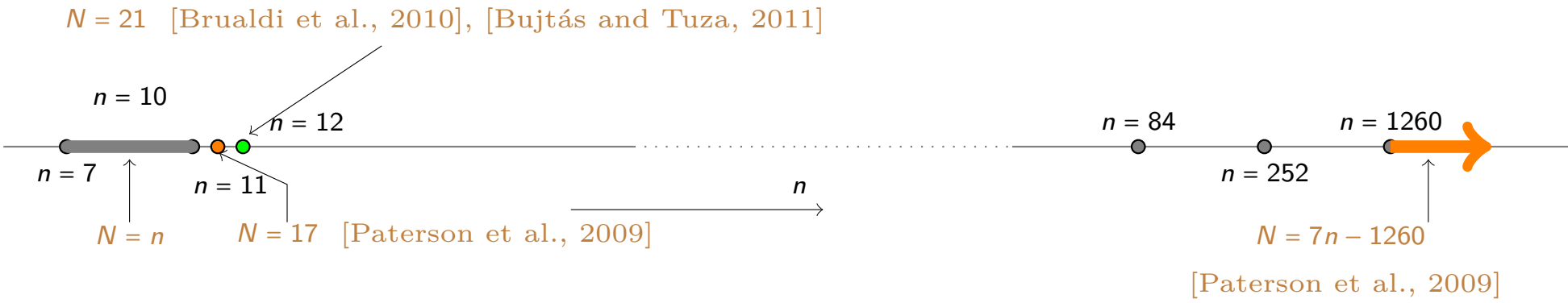
Present status

Optimal construction for $m = 10, k = 7$:



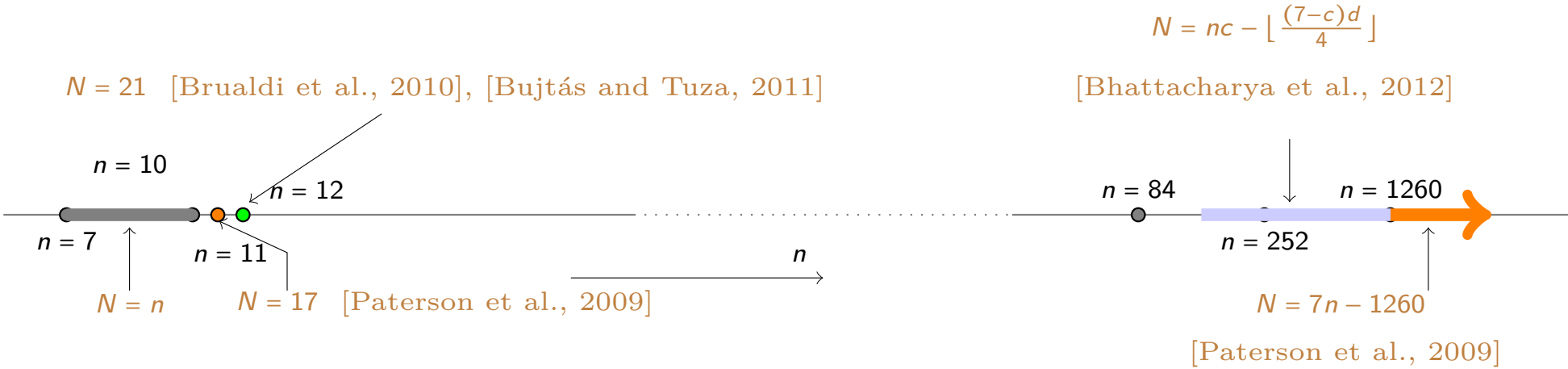
Present status

Optimal construction for $m = 10, k = 7$:



Present status

Optimal construction for $m = 10, k = 7$:



Outline of the talk

Problem

Concern c-uniform CBCs → each data item is stored in a fixed number (c) of servers. Given

Problem

Concern c-uniform CBCs → each data item is stored in a fixed number (c) of servers. Given

- Number of servers - m.

Problem

Concern c-uniform CBCs $\rightarrow$ each data item is stored in a fixed number (c) of servers. Given

- Number of servers - m.
- Degree of uniformity c.

Problem

Concern c-uniform CBCs $\rightarrow$ each data item is stored in a fixed number (c) of servers. Given

- Number of servers - m.
- Degree of uniformity c.
- Retrievability parameter - k.

Problem

Concern c-uniform CBCs $\rightarrow$ each data item is stored in a fixed number (c) of servers. Given

- Number of servers - m.
- Degree of uniformity c.
- Retrievability parameter - k.

Maximize the number of data items n, i. e., determination of $n(m, c, k)$.

Outline of the talk

[Paterson et al., 2009]

For any $c < k$, $n(m, c, k) \leq \frac{k-1}{\binom{k-1}{c}}\binom{m}{c}$..

[Paterson et al., 2009]

For any $c < k$, $n(m, c, k) \leq \frac{k-1}{\binom{k-1}{c}}\binom{m}{c}$..

[Ishai et al., 2004] $n(m, c, k) = \Omega(m^{c-1})$.

[Paterson et al., 2009] $n(m, c, k) = \Omega(m^{c-1+\frac{c}{k-1}})$.

} Non-constructive

[Paterson et al., 2009]
For any $c < k$, $n(m, c, k) \leq \frac{k-1}{\binom{k-1}{c}}\binom{m}{c}$..

[Ishai et al., 2004] $n(m, c, k) = \Omega(m^{c-1})$.

[Paterson et al., 2009] $n(m, c, k) = \Omega(m^{c-1+\frac{c}{k-1}})$.

} Non-constructive

Explicit constructions:

[Paterson et al., 2009]

For any $c < k$, $n(m, c, k) \leq \frac{k-1}{\binom{k-1}{c}}\binom{m}{c}$..

[Ishai et al., 2004] $n(m, c, k) = \Omega(m^{c-1})$.

[Paterson et al., 2009] $n(m, c, k) = \Omega(m^{c-1+\frac{c}{k-1}})$.

} Non-constructive

Explicit constructions:

[Paterson et al., 2009]

$$n(m, c, k) = \Theta(m^c) \text{ for } c = k-1, k-2.$$

[Paterson et al., 2009]
For any $c < k$, $n(m, c, k) \leq \frac{k-1}{\binom{k-1}{c}}\binom{m}{c}$..

[Ishai et al., 2004] $n(m, c, k) = \Omega(m^{c-1})$.

[Paterson et al., 2009] $n(m, c, k) = \Omega(m^{c-1+\frac{c}{k-1}})$.

} Non-constructive

Explicit constructions:

[Paterson et al., 2009]

$$n(m, c, k) = \Theta(m^c) \text{ for } c = k-1, k-2.$$

[Balachandran and Bhattacharya, 2012]

$$n(m, c, k) = \Theta(m^c) \text{ for } k - \lceil \log k \rceil \leq c \leq k-1$$

Balachandran, N. and Bhattacharya, S. (2012).
On an extremal hypergraph problem related to combinatorial batch codes.
submitted.

[Balachandran and Bhattacharya, 2012]

$$n(m, c, k) = o(m^c) \quad \text{for } c \leq k - 1 - \lceil \log k \rceil.$$

[Balachandran and Bhattacharya, 2012]

$$n(m, c, k) = o(m^c) \quad \text{for } c \leq k - 1 - \lceil \log k \rceil.$$

[Bujtás and Tuza, 2012]

Let $c \geq 2$, and $m \geq k \geq c + 1$. Then

$$n(m, c, k) = O(m^{c-1+\frac{1}{\lfloor \frac{k}{c+1} \rfloor}})$$

.

[Balachandran and Bhattacharya, 2012]

$$n(m, c, k) = o(m^c) \quad \text{for } c \leq k - 1 - \lceil \log k \rceil.$$

[Bujtás and Tuza, 2012]

Let $c \geq 2$, and $m \geq k \geq c + 1$. Then

$$n(m, c, k) = O(m^{c-1+\frac{1}{\lceil \frac{k}{c+1} \rceil}})$$

. In particular, if $c \leq \frac{k}{2} - 1$, this result improves the above result.

Balachandran, N. and Bhattacharya, S. (2012).
On an extremal hypergraph problem related to combinatorial batch codes.
submitted.

Bujtás, C. and Tuza, Z. (2012).
Turán numbers and batch codes.
manuscript.

Graph case ($c = 2$)

[Balachandran and Bhattacharya, 2012]

An exact result:

$n(m, 2, 5) = \lfloor \frac{m^2}{4} \rfloor$.

- bipartite graph with partite sets of size $\lceil \frac{m}{2} \rceil$ and $\lfloor \frac{m}{2} \rfloor$.

Graph case ($c = 2$)

[Balachandran and Bhattacharya, 2012]

An exact result:

$n(m, 2, 5) = \lfloor \frac{m^2}{4} \rfloor$.

- bipartite graph with partite sets of size $\lceil \frac{m}{2} \rceil$ and $\lfloor \frac{m}{2} \rfloor$.

An exact order of magnitude:

$n(m, 2, k) = \Theta(m^{3/2})$ for $k = 6, 7, 8$.

Graph case ($c = 2$)

[Balachandran and Bhattacharya, 2012]

An exact result:

$n(m, 2, 5) = \lfloor \frac{m^2}{4} \rfloor$.

- bipartite graph with partite sets of size $\lceil \frac{m}{2} \rceil$ and $\lfloor \frac{m}{2} \rfloor$.

An exact order of magnitude:

$n(m, 2, k) = \Theta(m^{3/2})$ for $k = 6, 7, 8$.

Lower bound:

Let $k \geq 8$ then

$$n(m, 2, k) = \begin{cases} \Omega(m^{1+\frac{2}{k-5}}) & \text{if } k = 5 \mod 6 \\ \Omega(m^{1+\frac{2}{k-4}}) & \text{if } k = 2 \mod 6 \text{ or } k = 4 \mod 6 \\ \Omega(m^{1+\frac{2}{k-3}}) & \text{if } k = 1 \mod 6 \text{ or } k = 3 \mod 6 \\ \Omega(m^{1+\frac{2}{k-2}}) & \text{if } k = 0 \mod 6 \end{cases}$$

for infinitely many values of n

Upper bound:

[Balachandran and Bhattacharya, 2012]

For $k \geq 4$,

$$n(m, 2, k) = O(m^{1+\beta})$$

where $\beta = \frac{1}{\lfloor \frac{k}{4} \rfloor}$.

Upper bound:

[Balachandran and Bhattacharya, 2012]

For $k \geq 4$,

$$n(m, 2, k) = O(m^{1+\beta})$$

where $\beta = \frac{1}{\lfloor \frac{k}{4} \rfloor}$. [Bujtás and Tuza, 2012]

$$n(m, 2, k) = O(m^{1+\alpha})$$

where $\alpha = \frac{1}{\lfloor \frac{k}{3} \rfloor}$.

Upper bound:

[Balachandran and Bhattacharya, 2012]

For $k \geq 4$,

$$n(m, 2, k) = O(m^{1+\beta})$$

where $\beta = \frac{1}{\lfloor \frac{k}{4} \rfloor}$. [Bujtás and Tuza, 2012]

$$n(m, 2, k) = O(m^{1+\alpha})$$

where $\alpha = \frac{1}{\lfloor \frac{k}{3} \rfloor}$.

[Balachandran and Bhattacharya, 2012]

$$n(m, 2, k) = \Theta(m^{\frac{4}{3}}), \text{ for } k = 9, 10, 11.$$

Thank You