S©MR ISSN 1813-3304

СИБИРСКИЕ ЭЛЕКТРОННЫЕ
МАТЕМАТИЧЕСКИЕ ИЗВЕСТИЯ

Siberian Electronic Mathematical Reports
http://semr.math.nsc.ru

Том 10, стр. 656–665 (2013) УДК 519.72
MSC 94A29

COMBINATORIAL VERSION OF THE SLEPIAN-WOLF CODING THEOREM FOR BINARY STRINGS

D.A. CHUMBALOV

Abstract. In this paper we study a combinatorial analogue of the Slepian-Wolf coding. We consider communication protocols with three parties (Alice, Bob, and Charlie). Alice and Bob hold binary strings X and Y respectively, of the same length n, with the Hamming distance between X and Y bounded by some threshold c. Alice and Bob send some messages to Charlie, and then Charlie should reconstruct both X and Y. The aim is to optimize communication complexity of a protocol, i.e., to minimize the lengths of messages sent by Alice and Bob.

We show that simple and most natural lower bounds for this problem give in fact the right answer – these bounds can be achieved by some (nontrivial) communication protocols. We consider two principal settings: (i) the Hamming distance between X and Y is an absolute constant c, and (ii) the Hamming distance between these strings is αn for some constant fraction α. In the first setting we propose a very simple lower bound and a deterministic, polynomial-time for all three participants communication protocol that asymptotically achieves this bound. This protocol is based on the checksums obtained from syndromes of the BCH codes. In the second setting we prove a nontrivial lower bounds for deterministic protocols. But the lower bounds for probabilistic protocols remain very simple, and we construct a protocol that asympotically achieves these simple lower bounds. In this probabilistic protocol we combine the technique of syndromes of linear codes with list-decoding and random hash-functions.

Keywords: Distributed source coding, Slepian-Wolf theorem, communication complexity.

Chumbalov, D.A., Combinatorial Version of the Slepian-Wolf Coding Theorem for Binary Strings.

Received July, 31, 2013, published December, 14, 2013.

1. Introduction

We consider the following communication problem with three participants: Alice, Bob and Charlie. Assume that Alice is given a binary string X and Bob is given a binary string Y of the same length n, and strings X and Y differ from each other in $c = c(n)$ positions. All three participants know the parameters n and c. Alice and Bob should send to Charlie some messages so that he can reconstruct both strings X and Y. No communication is possible between Alice and Bob, and no feedback can be sent from Charlie to Alice and Bob.

We study communication complexity of this problem. We say that this communication problem can be solved with messages of length r_A and r_B if there exists a communication *protocol* where Alice and Bob send to Charlie r_A and r_B bits respectively, and then Charlie reconstructs X and Y. The question is for which pairs (r_A, r_B) such a protocol exists. This problem is a combinatorial analogue of the classic Slepian-Wolf problem [1], which is widely investigated in the probabilistic setting. Let us note that a very special case of this problem for $r_B = \infty$ is well studied in communication complexity for deterministic as well as for probabilistic communication protocols, see [2] and a survey [3].

In this paper we focus on two main cases of the problem: (i) the distance between two strings c is an absolute constant that does not depend on n, and (ii) $c = \alpha n$ is a constant fraction of the length of strings. In both cases we prove very close upper and lower bounds for the set of implementable pairs (r_A, r_B):

- Case $c = const$, *deterministic protocols*
  1. Lower bound: for every communication protocol
     - (a) $r_A + r_B \geq n + \log_2(\sum_{k=0}^{c} C_n^k)$ and
     - (b) $r_A, r_B \geq \log_2(\sum_{k=0}^{c} C_n^k)$.
  2. Upper bound: if for some r_A, r_B
     - (a) $r_A + r_B = n + \log_2(\sum_{k=0}^{c} C_n^k) + d$ and
     - (b) $r_A, r_B \geq \log_2(\sum_{k=0}^{c} C_n^k)$,

     (where d is some constant depends on c) then the problem can be solved with messages of length r_A, r_B.

  To prove the upper bound above we use the similar method as was used in [4] and [5] for the problem in probabilistic formulation. For the sake of self-containedness we detail the construction in the *Section 2.2* and prove the bound in *Theorem 2*.
- Case $c = const$, *probabilistic protocols*
  1. Lower bound: for every $\epsilon < 1/2$ every probabilistic protocol satisfies
     - (a) $r_A + r_B \geq n + \log_2(\sum_{k=0}^{c} C_n^k) - 1$ and
     - (b) $r_A, r_B \geq \log_2(\sum_{k=0}^{c} C_n^k)$.
- Case $c = \alpha n$, *deterministic protocols*
  1. A simple lower bound for the general case: for every communication protocol
     - (a) $r_A + r_B \geq (1 + h(\alpha))n - o(n)$ and
     - (b) $r_A, r_B \geq h(\alpha)n - o(n)$,

     where $h(\alpha) = \alpha \log_2 \frac{1}{\alpha} + (1 - \alpha) \log_2 \frac{1}{1-\alpha}$.
  2. A nontrivial lower bound for a special case: for every communication protocol

$\forall\alpha \in (0, 1/4)\ \exists\beta > 0 : r_A = n\ \rightarrow\ r_B \geq (1+\beta)h(\alpha)n + o(n)$. It was proved by Orlitsky in [3].

- Case $c = \alpha n$, *probabilistic protocols*
  (1) Lower bound: for every $\epsilon < 1/2$ a communication protocol satisfies
    (a) $r_A + r_B \geq (1 + h(\alpha))n - o(n)$ and
    (b) $r_A, r_B \geq h(\alpha)n - o(n)$.
  (2) Upper bound: $\forall\epsilon > 0$ there exists a probabilistic protocol for our problem with an error less than ϵ and
    (a) $r_A + r_B = (1 + h(\alpha))n + o(n)$ and
    (b) $r_A, r_B \geq h(\alpha)n + o(n)$.

The last upper bound is the main result in this paper and it is formulated in *Section 3.3*. It combines the upgrading of the method (that is used in the upper bound for $c = const$) for the list-decoding codes and random hashing.

2. The distance between the strings is a constant

2.1. **Lower bound for deterministic protocols.** First of all, we formulate simple necessary conditions (i.e., a lower bound for the set of all implementable pairs (r_A, r_B)).

Theorem 1. *For every protocol where Alice sends r_A bits and Bob sends r_B bits, the following three inequalities are satisfied*

$$\begin{array}{cccl} (1) & r_A + r_B & \geq & n + \log_2(\sum_{k=0}^{c} C_n^k), \\ (2) & r_A, r_B & \geq & \log_2(\sum_{k=0}^{c} C_n^k). \end{array}$$

Proof. Let us introduce some notation. Let us fix some communication protocol for the problem under consideration. We assume that Alice and Bob send to Charlie messages $F_A(X)$ and $F_B(Y)$ respectively using some “coding"algorithms

$$F_A : \{0,1\}^n \rightarrow \{0,1\}^{r_A}\ ,\ F_B : \{0,1\}^n \rightarrow \{0,1\}^{r_B}\ ,$$

and Charlie “decodes"the received messages using some “decoding"algorithm

$$F_C : \{0,1\}^{r_B} \times \{0,1\}^{r_A} \rightarrow \{0,1\}^n \times \{0,1\}^n\ .$$

Let S be a set of n-bits strings (x, y) with Hamming distance at most c:

$$S = \{(x, y) \in \{0,1\}^n \times \{0,1\}^n \mid d(x, y) \leq c\}\ .$$

Then $|S| = 2^n \sum_{k=0}^{c} C_n^k$.

To be able to restore uniquely the initial pair (X, Y) from two messages r_A and r_B, the number of all possible pairs of messages must be at least $|S|$:

$$2^{r_A} 2^{r_B} \geq 2^n \sum_{k=0}^{c} C_n^k,$$

i.e.

$$r_A + r_B \geq n + \log_2(\sum_{k=0}^{c} C_n^k).$$

Further, for every Alice's string X Bob can be given one of $\sum_{k=0}^{c} C_n^k$ possible strings Y (any string at the distance at most c from X). Hence,

$$r_B \geq \log_2(\sum_{k=0}^{c} C_n^k).$$

Similarly,

$$r_A \geq \log_2(\sum_{k=0}^{c} C_n^k).$$

□

2.2. **Upper bound: a deterministic protocol.** It turns out that the simple lower bound formulated above is very close to an upper bound, i.e., to the sufficient conditions for (r_A, r_B), which guarantee that the required communication protocol exists:

Theorem 2. *For every c there exists a constant $d = d(c)$ such that our communication problem can be solved for all pairs (r_A, r_B) satisfying inequalities*

$$\begin{cases} r_A + r_B = n + \log_2(\sum_{k=0}^{c} C_n^k) + d, \\ r_A \geq \log_2(\sum_{k=0}^{c} C_n^k), \\ r_B \geq \log_2(\sum_{k=0}^{c} C_n^k). \end{cases}$$

Moreover, there exists a communication protocol with effective (deterministic, polynomial in time) algorithms for all three participants.

Let us sketch the construction of the protocol for Theorem 2. Alice and Bob send to Charlie some (clever chosen) bits of their strings X and Y, and in addition the syndromes of strings X and Y in a suitable linear error correcting code. We take the Hamming code for $c = 1$ and the BCH codes that correct c errors for $c > 1$ (standart information about coding theory can be found, e.g., in [7]). For $c = 1$ we achieve parameters that exactly match the lower bound from Theorem 1, so we obtain the complete solution of the problem (the necessary and sufficient conditions coincide). For $c > 1$ the gap between the bounds proven in Theorem 1 and Theorem 2 is only $O(1)$.

Proof. Let r_A, r_B be numbers that satisfy the conditions of the theorem (three linear inequalities). In what follows we describe the communication protocol. First we define the encoding rules for Alice and Bob, and then explain how Charlie can decode (X, Y) from the received messages.

We denote by H the parity-check matrix of the $[n,\ k,\ 2c+1]$ BCH code.

Alice

(1) Sends the first l (some integer from $[0, k]$ chosen by participants in advance) characters of the string X to Charlie (denote them X_1^l):

$$\underbrace{X_1 X_2 \dots X_l}_{X_1^l} X_{l+1} \dots X_n$$

(2) Computes the syndrome of string X in the BCH codes

$$h_A = H \cdot (X_1 \dots X_n)^T$$

and sends the result to Charlie.

Bob

(1) Transmits $k-l$ characters of the string Y starting with $(l{+}1)$-th to k-th (denote these bits Y_{l+1}^k):
$$Y_1Y_2\dots Y_l\underbrace{Y_{l+1}Y_{l+2}\dots Y_k}_{Y_{l+1}^k}Y_{k+1}\dots Y_n.$$

(2) Computes the syndrome of string Y in the BCH codes
$$h_B = H\cdot(Y_1\dots Y_n)^T$$
and sends the result to Charlie.

The total number of transmitted bits is
$$l+(n-k)+(k-l)+(n-k) = 2n-k = 2n-n+c\log_2 n+1 = n+c\log_2 n+1.$$
Furthemore, all the computations can be done by Alice and Bob in polynomial time.

Now we explain the most involved part of the protocol, the decoding algorithm for Charlie.

Charlie

(1) Receives two syndromes $h_A = HX$, $h_B = HY$ and k bits X_1^l, Y_{l+1}^k.
(2) Restores the error-pattern $E = X\oplus Y$ from the syndromes.
(3) Computes $X_{l+1}^k = Y_{l+1}^k\oplus E_{l+1}^k$.
(4) Computes $Y_1^l = X_1^l\oplus E_1^l$.
(5) Reconstructs strings X and Y.

First we show how to get the error-pattern $X\oplus Y$ from HX and HY. Since BCH code is a linear code, $HX\oplus HY = H(X\oplus Y) = HE$. We can easily find some word Z such that $HZ = HE$ by solving a system of linear equations (note that we find *some* Z, which can be different from E). Then, we apply to Z a standard decoding procedure for the BCH code and find a codeword $\hat{Z}$ (the unique codeword in c-neighborhood of Z). Next claim follows from general properties of linear codes.

Claim 1. *$\hat{Z}\oplus Z = E$.*

Proof. Suppose that $\hat{Z}\oplus Z = \hat{E}\neq E$. Then the weights (the number of 1's) of both words E and $\hat{E}$ are at most c. On the other hand,
$$H\hat{E} = H(\hat{Z}\oplus Z) = H\hat{Z}\oplus HZ = HZ = HE,$$
i.e., syndromes of E and $\hat{E}$ are equal to each other. It follows that
$$H(E\oplus\hat{E}) = 0.$$
That is, string $E\oplus\hat{E}$ is a codeword. But the weight of this string is at most $2c$. We get a contradiction with the assumption that this linear code corrects c errors. □

So, we have the error-pattern E. Now we explain how Charlie can recover X and Y.

Charlie computes $Y_{l+1}^k\oplus E_{l+1}^k$ and obtains X_{l+1}^k. Then, he concatenates it with received bits X_1^l and gets X_1^k.

W.l.o.g. we may assume that the first k symbols of codewords (of the BCH code) are the information bits, and the last $n-k$ symbols are the parity bits. So, given a syndrome HX and X_1^k, we can reconstruct the remaining bits of the word X by solving a system of linear equations $HX = h_A$. In a similar way Charlie reconstructs Y.

Note that all computations can be done in polynomial time. The only algorithmically nontrivial stage is decoding of the BCH code; but we can do it by the standard Berlekamp-Welch algorithm.

It remains to estimate the number of bits communicated in the constructed protocol. We need to show that the number of bits sent by Alice and Bob in this protocol matches the the claim of the theorem.

Claim 2. *$c\log_2 n = \log_2(\sum_{k=0}^{c} C_n^k) + d(c)$ where $d(c)$ is a constant depends on c.*

Proof.

$$\log_2(\sum_{k=0}^{c} C_n^k) \geq \log_2(C_n^c) \geq \log_2\left(\frac{n}{c}\right)^c \geq c\log_2 n + const(c)$$

$$\log_2(\sum_{k=0}^{c} C_n^k) \leq \log_2((c+1)C_n^c) \leq \log_2((c+1)\frac{n^c}{c!}) \leq c\log_2 n + const(c)$$

□

□

Now we got an effective, asymptotically optimal deterministic communication protocol. But if we allow randomness in protocols, can we improve our results by some *probabilistic* protocol? Assume that now the participants can toss (privately) random coins and use the randomness in the protocol; for all X and Y the result should be correct with probability at least $1-\epsilon$, for a small enough ϵ.

2.3. **Lower bound: probabilistic protocols.** Actually, replacement of deterministic protocols by probabilistic ones does not give us far better results:

Theorem 3. *For every $\epsilon < 1/2$ every probabilistic communication protocol satisfies*

(1) $r_A + r_B \geq n + \log_2(\sum_{k=0}^{c} C_n^k) - 1$ *and*
(2) $r_A, r_B \geq \log_2(\sum_{k=0}^{c} C_n^k)$.

Proof. We can prove these lower bounds even for the model with common randomness (when Alice and Bob have an access to a common source of random bits, Charlie is deterministic). In this model we may think of a probabilistic protocol as a probability distribution on a set P of *deterministic* protocols. Let us fix such a probabilistic protocol (such a distribution on deterministic protocols) and prove the lower bound for its communication complexity.

W.l.o.g. we assume that in all these protocols Alice and Bob send to Charlie r_A and r_B bits of information (communication complexities of all these deterministic protocols are the same, say, k). Let S be a set of all valid pairs of inputs:

$$S = \{(x, y) \in \{0,1\}^n \times \{0,1\}^n \mid d(x, y) \leq c\}.$$

$$|S| = 2^n \sum_{k=0}^{c} C_n^k.$$

The error of a probabilistic protocol must be less than an $\epsilon < 1/2$. Hence, for every pair (X, Y) a randomly chosen protocol with probability $1-\epsilon > 1/2$ contains this pair as a possible answer.

Hence, for a randomly chosen protocol P and a uniformly and randomly chosen valid pair (X, Y) with probability greater than $1/2$ we get a protocol that contains this pair of strings as a possible answer. It follows that some protocol in P contains more than $1/2$ valid pairs as answers, i.e. in that protocol there exist at least $|S|/2$ different possible answers. Obviously, in such a protocol Charlie can receive at least $|S|/2$ different pairs of messages from Alice and Bob. Thus, for this protocol

$$r_A + r_B \geq \log_2 k \geq \log_2 \frac{|S|}{2} \geq n + \log_2(\sum_{k=0}^{c} C_n^k) - 1.$$

A similar argument implies $r_A, r_B \geq \log_2(\sum_{k=0}^{c} C_n^k)$. □

3. The distance between the strings is a fraction of n

Now we assume that $c = \alpha n$, for some fixed $\alpha \in (0,\ 1/2)$.

3.1. **Lower bound for deterministic protocols.**

The bound from Theorem 1 is still valid, and (1) and (2) can be reformulated as

Theorem 4. *Let $h(\alpha) = \alpha \log_2 \frac{1}{\alpha} + (1-\alpha)\log_2 \frac{1}{1-\alpha}$. For every protocol where Alice sends r_A bits and Bob sends r_B bits, the following three inequalities are satisfied*

$$\begin{array}{lrcl} (1') & r_A + r_B & \geq & (1 + h(\alpha))n - o(n), \\ (2') & r_A, r_B & \geq & h(\alpha)n - o(n). \end{array}$$

Proof. If X and Y differ in αn positions, then, similarly to the proof of *Theorem 1*, we can obtain

$$\begin{array}{lrcl} (1') & r_A + r_B & \geq & n + \log_2(\sum_{k=0}^{\alpha n} C_n^k), \\ (2') & r_A, r_B & \geq & \log_2(\sum_{k=0}^{\alpha n} C_n^k). \end{array}$$

But $\sum_{k=0}^{\alpha n} C_n^k = 2^{h(\alpha)n + o(n)}$ (this can be easily proved using Stirling's approximation). Hence,

$$\begin{array}{lrcl} (1') & r_A + r_B & \geq & (1 + h(\alpha))n - o(n), \\ (2') & r_A, r_B & \geq & h(\alpha)n - o(n), \end{array}$$

□

In the contrast to the **first case** (when c is a constant), not all the points satisfying these bounds can be achieved by some (deterministic) protocol. For instance, the pair $(r_A, r_B) = (n, h(\alpha)n)$ is not achievable:

Proposition. *For every $\alpha \in (0, 1/4)$ there exists a $\beta > 0$ such that for all sufficiently large n and every deterministic communication protocol for our problem with $r_A = n$ it holds that $r_B \geq (1 + \beta)h(\alpha)n$.*

(This proposition follows from a result of Orlitsky in [2], *Theorem 2.*)

Since deterministic protocols cannot achieve all pairs (r_A, r_B) that satisfy the natural necessary conditions, we try probabilistic protocols.

In this case, on the one hand, randomness once again cannot help too much: a counterpart of Theorem 1 is still valid, but on the other hand, the randomness is somewhat useful: we construct a communication protocol with parameters that are asymptoticly close to $(1')$ and $(2')$.

3.2. **Lower bound for probabilistic protocols.**

Theorem 5. *For randomized protocols communication complexity cannot be far below the standard lower bounds. More precisely, for every $\epsilon < 1/2$ the communication complexity of a probabilistic protocol for the problem in case $\alpha \in (0, 1/2)$ satisfies the following three inequalities*

$$\begin{array}{lrcl} (1') & r_A + r_B & \geq & (1 + h(\alpha))n - o(n), \\ (2') & r_A, r_B & \geq & h(\alpha)n - o(n), \end{array}$$

Proof. Similarly to the proof of *Theorem 3.* □

3.3. **Upper bound: a probabilistic protocol.**

First let us formulate the following lemma (the proof can be found, e.g., in [3]):

Lemma 1. *Let $U = \{1, ..., 2^n\}$ and $S \subseteq U$: $|S| = k$. Then for any $a \in \mathbb{N}$ there exists a number $A = ank^2 \log_2(ank^2) + o(ank^2)$ such that a uniformly chosen prime number q from $[1, A]$ with probability $1 - \frac{1}{a}$ set the function $f_q : x \to x \bmod q$ with no collisions on the S.*

Moreover, it takes $O(n^2)$ to compute $f_q(x)$ and $O(\frac{A}{\log_2 \log_2(A)})$ to chose a prime number.

Theorem 6. *For every $\epsilon > 0$, for $c = \alpha n$, and all pairs (r_A, r_B) satisfying inequalities*

$$r_A + r_B \geq (1 + h(\alpha))n + o(n) \text{ and } r_A, r_B \geq h(\alpha)n + o(n),$$

there exists a probabilistic protocol for our problem (with probability of error $2\epsilon - \epsilon^2$).

The proof of this theorem is the most involved construction in this paper. It combines the technique from Theorem 2 with the idea of list decoding and random hashing, similar to [3]. Loosely speaking, now instead of codes correcting c errors (the trick in Theorem 2) we take a code with list-decoding for radius c. Then, our asymptotical bounds come from the list-decoding capacity theorem (see, e.g., [6]). In the marginal case $r_A = n$ our construction coincides with the protocol proposed by Chuklin in [3].

Proof. According to the list-decoding capacity theorem, there exists a linear $(\alpha, L \geq 1)$-list decodable code $\mathcal{C} \subseteq \Sigma^n$, if its rate $R = \frac{k}{n} = 1 - h(\alpha) - 1/L$, where by k we denote the number of information symbols in codewords from $\mathcal{C}$. Let us fix such a code and its parity-check matrix H.

New probabilistic protocol is like the old deterministic one that we used in *Theorem 2*, but it includes some extra operations:

Alice

(1) Sends the first l (once again some chosen in advance by Alice, Bob and Charlie integer from $[0, k]$) characters of the string X to Charlie (denote them X_1^l):

$$\underbrace{X_1 X_2 \dots X_l}_{X_1^l} X_{l+1} \dots X_n$$

(2) Computes the syndrome of the string X in $\mathcal{C}$

$$h_A = H \cdot (X_1 \dots X_n)^T$$

and sends the result to Charlie.

(3) Uniformly choses some random prime number q_1 from a segment $[1, D]$ (an integer D participants chose in advance according to ϵ: $D = \frac{nL^2}{\epsilon} \log_2 \frac{nL^2}{\epsilon} + o(\frac{nL^2}{\epsilon})$) and sends it to Charlie.

(4) Sends to Charlie the result of hashing $f_{q_1}[X] = X \; mod \; q_1$.

Bob

(1) Transmits $k - l$ characters of the string Y starting with $(l{+}1)$-th to k-th (denote these bits Y_{l+1}^k):

$$Y_1 Y_2 \ldots Y_l \underbrace{Y_{l+1} Y_{l+2} \ldots Y_k}_{Y_{l+1}^k} Y_{k+1} \ldots Y_n.$$

(2) Computes the syndrome of the string Y in in $\mathcal{C}$

$$h_B = H \cdot (Y_1 \ldots Y_n)^T$$

and sends the result to Charlie.

(3) Uniformly choses some random prime number q_2 from a segment $[1, D]$ and sends it to Charlie.

(4) Sends to Charlie the result of hashing $f_{q_2}[Y] = Y \; mod \; q_2$.

The total number of transmitted bits is:

$$l + (1 - R)n + 2\log_2 \frac{nL^2}{\epsilon} + (k - l) + (1 - R)n + 2\log_2 \frac{nL^2}{\epsilon} + o(n) =$$

$$= 2(1 - R)n + Rn = (1 + h(\alpha))n + \frac{1}{L} + o(n) = (1 + h(\alpha))n + o(n).$$

Charlie

(1) Receives two syndromes $h_A = HX$, $h_B = HY$, k bits X_1^l, Y_{l+1}^k, two primes q_1, q_2 and hash sums $f_{q_1}[X]$ and $f_{q_2}[Y]$.

(2) Restores the error-pattern $E = X \oplus Y$.

(3) Computes $X_{l+1}^k = Y_{l+1}^k \oplus E_{l+1}^k$.

(4) Computes $Y_1^l = X_1^l \oplus E_1^l$.

(5) Reconstructs strings X and Y.

Now Charlie uses the same algorithm of getting the error-pattern $E = X \oplus Y$, but in this protocol due to using list-decodable linear code as a base code, he will get a list of L candidates, containing the correct one. Then for every $\hat{E}$ from that list Charlie reconstructs a pair of strings $(\hat{X}, \hat{Y})$ and add it to a new list of pairs-candidates. At the end he gaines a list of L pairs-condidates, containing a correct pair (X, Y). Next Charlie applies hash-functions f (based on q_1 and q_2) to every pair (X', Y') from the second list, getting the third list – a list of pairs $(f_{q_1}[X'], f_{q_2}[Y'])$ and mathes each element with the recieved pair of hashes $(f_{q_1}[X], f_{q_2}[Y])$. When some pair from that list mathes the recieved one, Charlie takes the corresponding pair from the second list (pairs-candidates) as the answer. According to Lemma 1 the probability of an error is $2\epsilon - \epsilon^2$, i.e., with probability $(1 - \epsilon)^2$ the uniformly chosen q_1 and q_2 set non-collision hash-functions f_{q_1} and f_{q_2} and Charlie is able to uniquely restore the correct pair (X, Y).

□

The disadvantage of this protocol is that the participants need to perform exponentially long computations (there is no explicit constructions of codes achieving the list-decoding capacity).

4. Open questions

(1) Can we generalize *Proposition* and prove the following statement: For every $\alpha \in (0, 1/4)$ there exists a $\beta > 0$ such that for every deterministic communication protocol for our problem it holds $r_A + r_B \geq n + (1 + \beta)h(\alpha)n + o(n)$.

(2) Can we achieve the communication complexities from *Theorem 6* with an *effective* (randomized) protocol, where Alice, Bob and Charlie need only polynomial time computations?

5. Acknowledgements

The author would like to thank Andrei Romashchenko for posing the problem, fruitful discussions and his help in writing this paper.

References

[1] D. Slepian and J. K. Wolf, *Noiseless Coding of Correlated Information Sources*, IEEE Transactions on Information Theory, **19** (1973), 471–480. MR0421858

[2] A. Orlitsky, K. Viswanathan *One-Way Communication and Error-Correcting Codes*, IEEE Transactions on Information Theory, **49**, No. 7 (2003), 1781–1788. MR1985578

[3] A. Chuklin *Effective protocols for low-distance file synchronization*, arXiv:1102.4712, 2011, *in Russian.*

[4] N. Gehrig and P. L. Dragotti, *Symmetric and asymmetric Slepian-Wolf codes with systematic and nonsystematic linear codes*, IEEE Communications Letters, **9**, No. 1 (2005), 61–63.

[5] P. Tan and J. Li, *A practical and optimal symmetric Slepian-Wolf compression strategy using syndrome formers and inverse syndrome formers*, Proceeding of the 43rd Annual Allerton Conference on Communication, Control and Computing, 2005.

[6] V. Guruswami, J. Hastad, M. Sudan and D. Zuckerman, *Combinatorial bounds for list decoding*, IEEE Transactions on Information Theory, **48**, No. 5 (2002), 1021–1035. MR1907395

[7] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland, 1977. MR0465509 MR0465510

Daniyar Asetovich Chumbalov
Moscow Institute of Physics and Technology,
9 Institutskiy per.,
Dolgoprudny, 141700, Russian Federation
E-mail address: keimperk@gmail.com