Individual Communication Complexity

Harry Buhrman [*] Hartmut Klauck [†] Nikolai Vereshchagin [‡] Paul Vitányi [§]

Abstract

We initiate the theory of communication complexity of individual inputs held by the agents, rather than worst-case or average-case. We consider total, partial, and partially correct protocols, one-way versus two-way, with and without help bits.

1 Introduction

Assume Alice has input x and Bob has input y and they want to compute a function $f(x, y)$ by communicating information and local computation according to a fixed protocol $P = (P_A, P_B)$ where P_A is the protocol executed by Alice, and P_B is the protocol executed by Bob. For definiteness assume that the requirement is that Alice outputs $f(x, y)$. We are only interested in minimizing the number of bits communicated between Alice and Bob as in [9, 4]. In the usual setting one considers the worst-case or average-case over all inputs x, y of given length n. However, in current situations like replicated file systems, and cache coherence algorithms, in multiprocessor systems and computer networks, the worst-case or average-case are not necessarily significant. The files or updates can be very large; but in real life they may typically be non-random and have considerable regularities or correlations that allow the communicated information to be greatly compressible. Neither the worst-case not the average-case may be relevant; one wants to analyze the individual case. This gives also much more information: from the individual case-analysis one can easily derive the worst-case and the average-case, but not the other way around. Indeed, certain phenomena have no counterpart in more traditional settings: For example, there are inputs for Bob such that irrespective of Alice's input, every "simple" total protocol requires arbitrarily higher communication complexity than some more "complex" total protocol. Our results are expressed in terms of Kolmogorov complexity [6], the minimal number of bits from which the data can be decompressed by effective computation. We use the "plain" Kolmogorov complexity denoted as $C(x), C(x|y), C(x, y)$ for the absolute complexity of x, the conditional complexity of x given y, and the joint complexity of x, y. Increased compression of the data approximates the Kolmogorov complexity more and more, but the actual value is uncomputable in general. Given x, y, and assuming that Alice and Bob have a protocol P that works correctly on x, y, we study the *individual communication complexity* $CC^P(x, y)$ defined as the number of bits Alice with input x and Bob with input y exchange using protocol P. We refer to a standard definition of communication protocol [4]. We assume that the protocol identifies the length n of the strings on which it works. By the complexity of a protocol P we mean its plain Kolmogorov complexity conditional to n, denoted as $C(P|n)$.

Results and Related Work: We use the framework of communication complexity as in [9, 4]. As far as we are aware there is no previous work on individual communication complexity. We formulate a theory of individual communication complexity, and first analyze the "mother" problem, the indentity function, where Alice outputs the input of Bob. We look at special functions such as the inner product, random functions, and the equality function. We then turn to the question of analyzing the communication complexity, with respect to the best protocol of given complexity, for the mother problem (identity function). For total protocols that are always correct, the power of one-way protocols equals that of two-way protocols, but for partially correct protocols or partial protocols, two-way protocols are remarkably more powerful. We establish a relation with Kolmogorov's Structure function, and the existence of strange "non-communicable" inputs of possibly low Kolmogorov complexity for total protocols—for which the communication complexity of every total protocol

[*] CWI and University of Amsterdam
[†] IAS, Princeton
[‡] Moscow State University
[§] CWI and University of Amsterdam

is necessarily very large (almost the literal uncompressed input needs to be communicated) unless all of the input is hard-wired in the protocol. It is shown that for partial protocols two-way is more powerful than one-way when we use help bits.

2 The mother function: Identity

We start with listing some easy facts that establish lower and upper bounds on individual communication complexity with respect to individual protocols P expressed in terms of $C(y|n)$, $C(y|P)$ and compared to $C(y|x)$. We assume that the protocols do not depend on x, y, they are uniform, and they compute the function concerned on strings of length n. Let C be a constant such that $C(y|n) \leq n + C$ for all y.

Let $I(x, y) = y$ be the identity function: Alice with input x and Bob with input y compute output y by Alice. This is the "mother" function: for if Alice can compute I then she can compute every computable function f.

(1) For all n there is a protocol P of complexity $n + O(1)$ to compute the identity function such that for all x, y of length n we have $CC_I^P(x, y) \leq C(y|n)$.
Indeed, assume Bob knows $L_n = |\{p \mid |p| \leq n + C,\ U(p) \text{ halts}\}|$. ($U$ is the reference universal Turing machine.) Then Bob can find all halting programs of length at most $n + C$ by enumerating them until he obtains L_n halting programs. This allows him to find a shortest program y^* for y. He transmits that program to Alice and Alice computes y. The complexity of this protocol is $C(L_n) + O(1) = n + O(1)$.

(2) The complexity bound $n + O(1)$ on $C(P|n)$ in item (1) is tight. For every protocol of complexity less than n the assertion of item (1) is false: for all P there are x, y such that $CC_I^P(x, y) \geq n$ but $C(y|P) = O(1)$ (and hence $C(y|n) \leq C(P|n) + O(1)$, that is $C(y|n)$ is much smaller than $CC_I^P(x, y)$ if $C(P|n)$ is much smaller than n).
Indeed, let ϵ be the empty string and let y be the first string such that $CC_I^P(\epsilon, y) \geq n$ (by counting arguments there is such y).

(3) For every protocol P to compute identity function and for every x, y we have $CC_I^P(x, y) \geq C(y|P) - O(1)$.
Let c be the conversation between Alice and Bob on inputs x, y. It suffices to prove that given P, c we can find y. It is known [4] that the set of all pairs (x', y') such that the conversation between Alice and Bob on input (x', y') is equal to c is a rectangle, that is, has the form $X \times Y$, for some $X, Y \subset \{0,1\}^n$. The set Y is a one-element set, as for every $y' \in Y$ Alice outputs y also on the input (x, y') (the output of Alice depends on c, P, x only). We can find Y given P, c and since $Y = \{y\}$ we are done.

By item (2), for every protocol there are x, y such that the right hand side of the inequality $CC_I^P(x, y) \geq C(y|P) - O(1)$ is much less than its left hand side, more specifically, $C(y|P) = O(1)$ and $CC_I^P(x, y) \geq n$.

(4) How is $CC_I^P(x, y)$ related to $C(y|x)$? By item (3) we have $CC_I^P(x, y) \geq C(y|x) - C(P) - O(\log C(P))$ for all x, y. For all P this inequality is not tight for some x, y: there are x, y such that $C(y|x) = O(1)$ but $CC_I^P(x, y) \geq n$.
Indeed, let $x = y$. We need to prove that for some x it holds $CC_I^P(x, x) \geq n$. For every x let $c(x)$ denote the conversation on the pair (x, x). For every x the set of input pairs (x', y') producing the conversation $c(x)$ is a rectangle of height 1, as we have seen in item (3). Therefore $c(x)$ are pairwise different for different x hence for some x we have $|c(x)| \geq n$.

(5) However, for some P, x, y the inequality $CC_I^P(x, y) \geq C(y|x) - C(P|n) - O(\log C(P|n))$ is close to an equality: for all α there are P, x, y such that $CC_I^P(x, y) = C(y|x) - \alpha + O(1)$ and $C(P|n) \leq \alpha + O(1)$.
Indeed, let x be some string. Let y be a random string of length n, independent of x, that is, $C(y|x) = n + O(1)$. Let P be the following protocol: Bob first looks whether his string y' has the same prefix of length α than y. If this is the case he sends to Alice 0 and then $n - \alpha$ remaining bits of y' and Alice prefixes the $n - \alpha$ received bits by α first bits of y and outputs the resulting string. Otherwise Bob sends to Alice 1 and then y'. The complexity of this protocol is at most $\alpha + O(1)$, as both Alice and Bob need to know only the first α bits of y. And we have $CC_I^P(x, y) = n - \alpha = C(y|x) - \alpha + O(1)$.

3 Other functions

Because Alice can compute every computable function once she knows Bob's input, we have $CC_f^{P'}(x, y) \leq CC_I^P(x, y)$, with $C(P') \leq C(P, f) + O(1)$.

The trivial lower bound on the individual communication complexity of a function f is $CC^P(x,y) \geq C(f(x,y) \mid x, P) - O(1)$ (and hence $TCC^{\alpha}_f(x,y) \geq C(f(x,y) \mid x) - \alpha - O(\log \alpha)$ anticipating on a later defined notion). In this section we establish some nontrivial lower bounds on $CC^P(x,y)$ for P computing f on all arguments for the inner product function, the equality function and for random Boolean functions.

3.1 Inner Product

We extend an argument introduced in [2]. Initially, Alice has a string $x = x_1, \ldots, x_n$ and Bob has a string $y = y_1, \ldots, y_n$ with $x, y \in \{0,1\}^n$. Alice and Bob compute the inner product of x and y modulo 2

$$f(x,y) = \sum_{i=1}^{n} x_i \cdot y_i \bmod 2$$

with Alice ending up with the result.

Theorem 1. *Every deterministic protocol P_f computing the inner product function f (without help bits) requires at least $CC^P(x,y) \geq C(x,y \mid P) - n - O(1)$ bits of communication on all x, y.*

Proof. Fix a communication protocol P that computes the inner product. Let Alice's input be $x = x_1 \ldots x_n$ and Bob's input be $y_1 \ldots y_n$. Run the communication protocol P on x, y and let $c(x,y)$ be the communication between Alice and Bob. Note that P can be viewed as a tree with $c(x,y)$ a path in that tree [4]. Hence $c(x,y)$ form a prefix free set. Consider the set $S = S(x,y)$ defined by

$$S := \{(a,b) \mid C(a,b) = C(x,y), \text{ and Alice outputs } f(x,y) \text{ having the conversation } C(x,y) \text{ and input } a\}.$$

We claim that $|S| \leq 2^n$. To prove the claim assume first that $f(x,y) = 0$. Let X be the first projection X of S and Y be the second projection of S. Being an intersection of two rectangles, S is a rectangle too. As P computes f we know that $f(a,b) = 0$ for all $(a,b) \in S$. In other words, every element of X is orthogonal to every element in Y hence $\text{rank}(X) + \text{rank}(Y) \leq n$. Thus $|S| = |X| \cdot |Y| \leq 2^{\text{rank}(X) + \text{rank}(Y)} \leq 2^n$. Assume now that that $f(x,y) = 1$. Again $S = X \times Y$ for some X, Y and $f(a,b) = 1$ for all $(a,b) \in S$. Subtracting x from the first component of all pairs in S we obtain a rectangle S' such that $f(a,b) = 0$ for all $(a,b) \in S'$. By above argument, we have $|S'| \leq 2^n$. As $|S'| = |S|$ we are done.

Given P, $C(x,y)$, $f(x,y)$ and the index of (x,y) in S we can compute (x,y). By the prefix free property, $C(x,y)$ and the index of (x,y) can be concatenated without delimiters. Consequently, $C(x,y|P) \leq |c(x,y)| + n + O(1)$. □

REMARK 1. The result of the theorem is only significant for $C(x,y) > n$, but it cannot be improved. Namely, $CC^P(x,y) \geq C(x,y \mid P) - n - O(1) = C(y \mid x, P) - (n - C(x \mid P)) - O(\log n)$, where the equality follows from the "symmetry of information" property of Kolmogorov complexity [6]. The term $n - C(x \mid P)$ is a called the *randomness deficiency* of x with respect to P. Clearly, for $x = 00 \ldots 0$ it is maximal with $C(x \mid P) = O(\log n)$ and Alice knows already from her input x that $f(x,y) = 0$ and no bits, or only one bit, depending on the protocol conventions, need to be exchanged: $CC^P_f(x,y) = 0$ irrespective of the complexity of y which can be n.

3.2 Random Functions

Alice has $x = x_1 \ldots x_n$ and Bob has $y_1 \ldots y_n$, and $f : \{0,1\}^{2n} \to \{0,1\}$ satisfies

$$C(f \mid n) \geq 2^{2n} - n. \tag{1}$$

The latter condition means that the truth table describing the outcomes of f for the 2^n possible inputs x (the rows) and the 2^n possible inputs for y (the columns) has high Kolmogorov complexity. If we flip the truth table for a prospective f using a fair coin, then with probability at least $1 - 2^{-n}$ it will satisfy (1).

Theorem 2. *Every deterministic protocol P computing a function f satisfying (1) (without help bits) requires at least $CC^P_f(x,y) \geq \min\{C(x \mid P), C(y \mid P)\} - \log n - O(1)$.*

Proof. Run the communication protocol P on x, y and let $c(x,y)$ be the communication between Alice and Bob. Consider the set $S = S(x,y)$ defined by

$$S = \{(x', y') \mid c(x', y') = c(x,y), \text{ and Alice outputs } f(x,y) \text{ having the conversation } c(x,y) \text{ and input } x'\}.$$

Then S is a *monochromatic rectangle* in the function table of f (that is, $f(x', y') = f(x,y)$ for all $(x', y') \in S$). Suppose the rectangle S has dimensions $a \times b$. Then we can describe f by giving $f(x,y)$, the value of a in $2\log a + O(1)$ bits, the value of b in $2\log b + O(1)$ bits, the positions of the rows of the rectangles an bits, the positions of the columns of the rectangles in bn bits, all of the table except the rectangle, in row-major order, in $2^{2n} - ab$ bits. This description must have length at least the Kolmogorov complexity, so by (1) we find

$$2^{2n} - ab + (a+b)n + 2\log ab + O(1) \geq 2^{2n} - n.$$

Assume w.l.o.g. that $b \geq a$. Then $a < 3n$ if n is large enough. (Otherwise we would have $3bn \leq ab \leq (2b+1)n + 2\log b^2 + O(1)$.) Given the communication sequence $c(x,y)$, n and $f(x,y)$ we can find the rectangle S that it defines. Then, we can reconstruct x by indicating its row in the rectangle. Then $C(x \mid P) \leq |c(x,y)| + \log n + O(1)$. □

3.3 Equality and Functions with Large Monochromatic Rectangles

Let f be the equality function, with $f(x,y) = 1$ if $x = y$ and 0 otherwise.

Theorem 3. *For every deterministic protocol P computing f we have $CC^P(x,x) \geq C(x \mid P) - O(1)$ for all x, y. On the other hand there is P of complexity $O(1)$ such that there are x, y ($x \neq y$) with $C(x \mid P), C(y \mid P) \geq n - 1$ for which $CC_f^P(x,y) = 2$.*

Proof. Lower bound: Since trivially the communication sequence must be different and uniquely identify x if both Alice and Bob have input x, we have also $CC_f^P(x,x) + O(1) \geq C(x \mid P)$.

Upper bound: In the function table the lower left rectangle consisting of all x's beginning with 0 and all y's beginning with 1 is monochromatic (entries are all 0). Thus, a protocol where Bob communicates one bit to Alice indicating whether x starts with 0 allows Alice, in case y starts with 1, to output 0. Otherwise Alice and Bob start the default protocol. Thus, for such x, y and P we have $CC_f^P(x,y) = 2$. By simple counting for some such inputs we have $C(x \mid P), C(y \mid P) \geq n - 1$. □

Generalizing this idea, every function that contains large monochromatic rectangles, of size say $2^{2n}/n^{O(1)}$, has many pairs x, y of complexity close to n for which the individual communication complexity drops to $O(\log n)$, as follows:

In round 1 Bob tells Alice in which large rectangle (if any) his input is situated, by sending the index of the rectangle to Alice, and 0 otherwise. If Bob did send an index, and Alice's input is in that rectangle as well, then Alice outputs the color ("0" or "1") of the rectangle. Otherwise, Alice starts a default protocol.

4 Total protocols

Let f be a function defined on pairs of strings of the same length. Assume that Alice has x, Bob has y and Alice wants to compute $f(x,y)$. As the complexity measure we consider the number of bits communicated between Alice and Bob. The naive definition of the individual communication complexity of the value of the function f on the argument (x,y) is the number of communicated bits in the "best" communication protocol. Then, for every x, y there is a protocol with no communication at all on (x,y): the string y is hard wired into the protocol. To meaningfully capture the individual communication complexity of computing a function $f(x,y)$ we define now the following notion.

Definition 1. Let α be a natural number parameter. Let $TCC_f^\alpha(x,y)$ stand for the minimum $CC^P(x,y)$ over all total protocols P of complexity at most α that always compute f correctly (being total such a protocol terminates for all inputs, and not only for (x,y)).

For $\alpha = n + O(1)$ we have $TCC_f^\alpha(x,y) = 0$ for all computable f and all x, y, since we can hard wire y into the protocol. Therefore it is natural to consider only α that are much smaller than n, say $\alpha = O(\log n)$. Since computation of the Identity function suffices to compute all other (recursive) functions we have $TCC_f^{\alpha+O(1)}(x,y) \leq TCC_I^\alpha(x,y)$. The trivial lower bound is $TCC_f^\alpha(x,y) \geq C(f(x,y) \mid x) - \alpha - O(\log \alpha)$. For $f = I$ this gives $TCC_I^\alpha(x,y) \geq C(y \mid x) - \alpha - O(\log \alpha)$.

4.1 One-way equals two-way for Identity

Let $TCC^{\alpha}_{f,\text{1-way}}(x,y)$ stand for the minimum $TCC^P(x,y)$ over all one-way (Bob sends a message to Alice) total protocols P of complexity at most α computing f (over all inputs, and not only on (x,y)). It is clear that $TCC^{\alpha}_{f,\text{1-way}}(x,y)$ does not depend on x: indeed, consider for given (x,y) the best protocol P; that protocol sends the same message on every other pair (x',y) hence $TCC^{\alpha}_{f,\text{1-way}}(x',y) \le TCC^{\alpha}_{f,\text{1-way}}(x,y)$. Therefore we will use the notation $TCC^{\alpha}_f(y)$ dropping both x and "1-way". Obviously,

$$TCC^{\alpha}_f(x,y) \le TCC^{\alpha}_f(y)$$

for all α, x, y, f.

Surprisingly, for $f = I$, the Identity function, this inequality is an equality. That is, for total protocols "1-way" is as powerful as "many-way." More specifically, the following holds.

Theorem 4. *There is a constant C such that for all α, x, y we have*

$$TCC^{\alpha+C}_I(y) \le TCC^{\alpha}_I(x,y).$$

Proof. Pick a two-way protocol P witnessing $TCC^{\alpha}_I(x,y) = l$. Let $c = c(x,y)$ be the conversation according to P between Alice and Bob on inputs x, y. It is known that the set of all pairs (x',y') such that the conversation between Alice and Bob on input (x',y') is equal to c is a rectangle, that is, has the form $X \times Y$, for some $X, Y \subset \{0,1\}^n$. The set Y is a one-element set, as for every $y' \in Y$ Alice outputs y also on the input (x, y') (the output of Alice depends on c, P, x only).

Consider the following 1-way protocol P': find an x' with minimum $c(x',y)$ and send $c(x',y)$ to Alice. Alice then finds the set of all pairs (x'',y') such that the conversation between Alice and Bob on input (x'',y') is equal to $c(x',y)$. As we have seen that set has the form $X \times \{y\}$ for some X. Thus Alice knows y. As $|c(x',y)| \le |c(x,y)| = TCC^{\alpha}_I(x,y)$ and $C(P'|P) = O(1)$ we are done. □

4.2 Non-Communicable objects

The function $TCC^{\alpha}_I(y)$, as a function of y, α, essentially coincides with *Kolmogorov structure function* $h_y(\alpha)$ studied in [3, 8]. The latter is defined by

$$h_y(\alpha) = \min_S \{\log |S| : S \ni y,\ C(S) \le \alpha\},$$

where S is a finite set and $C(S)$ is the length (number of bits) in the shortest binary program from which the reference universal machine U computes a listing of the elements of S and then halts. More specifically we have

$$\begin{aligned} TCC^{\alpha+O(1)}_I(y) &\le h_y(\alpha), \\ h_y(\alpha + O(\log n)) &\le TCC^{\alpha}_I(y). \end{aligned} \tag{2}$$

To prove the first inequality we have to transform a finite set $S \ni y$ into a one-way protocol P of complexity at most $\alpha = C(S) + O(1)$ witnessing $TCC^{\alpha}_I(y) \le \log |S|$. The protocol just sends the index of y in S, or y literally if $y \notin S$.

To prove the second inequality we have to transform a one-way total protocol P into a finite set $S \ni y$ of complexity at most $C(P) + O(\log n)$ with $\log |S| \le CC^P(y)$. The set consists of all y' on which P sends the message of the same length l as the length of the message on y. Obviously, $|S| \le 2^l = 2^{CC^P(y)}$ and to specify S we need a program describing P and l. Thus $C(S) \le C(P) + O(\log TCC^{\alpha}_I(y)) \le C(P) + O(\log n)$.

For the properties of $h_y(\alpha)$, which by Theorem 4 are also properties of $TCC^{\alpha}_I(x,y)$, its relation with Kolmogorov complexity $C(y)$ of y and possible shapes of the function $\alpha \mapsto h_y(\alpha)$ we refer to [8].

We will present here only a few properties. First, two easy inequalities: For all $\alpha \ge O(1)$ and all x, y we have

$$C(y|n) - \alpha - O(\log \alpha) \le TCC^{\alpha}_I(y) \le n - \alpha + O(1). \tag{3}$$

The first inequality is the direct consequence of the inequality $C(y|n) \le CC^P(y) + C(P|n) + O(\log C(P|n))$, which is trivial. To prove the second one consider the protocol that sends $n - \alpha + C$ bits of y (for appropriate constant C) and the remaining α bits are hardwired into the protocol. Its complexity is at most $\alpha - C + O(1) \le \alpha$ for appropriate choice of C.

The second property is not so easy. Given y, consider values of α such that

$$TCC_I^{\alpha}(y) + \alpha = C(y) + O(1). \tag{4}$$

That is, the protocol P witnessing (4) together with the one-way communication record Bob sends to Alice form a two-part code for y that is—up to an independent additive constant—as concise as the shortest one-part code for y (that has length $C(y)$ by definition). Following the usage in [8] we call P a "sufficient" protocol for y. The descriptions of the protocol plus the communication precisely describe y, and in fact, it can be shown that the converse holds as well (up to a constant additive term). There always exists such a protocol, since the protocol that contains y hard wired in the form of a shortest program of length $C(y)$ satisfies the equality with $\alpha = C(y)+O(1)$ and $TCC_I^{\alpha}(y) = 0$. By definition we cannot have $TCC_I^{\alpha}(y)+\alpha < C(y)-O(1)$, but for α sufficiently small we have $TCC_I^{\alpha}(y)+\alpha > C(y)+O(1)$. In fact, for every form of function satisfying the obvious constraints on TCC_I^{α} there is a y such that $TCC_I^{\alpha}(y)$ realizes that function up to logarithmic precision. This shows that there are essentially non-communicable strings. More precisely:

Theorem 5. *For every $k \le n$ and monotonic decreasing function $h(\alpha)$ on integer domain $[0,k]$ with $h(0) = n$, $h(k) = 0$, $C(h) = O(\log n)$, and $h(\alpha) + \alpha \ge k$ for $\alpha \in [0,k]$, there is a string y of length n and $C(y) = k$ such that*

$$\begin{aligned} TCC_I^{\alpha+O(\log n)}(y) &\le h(\alpha), \\ h(\alpha + O(\log n)) &\le TCC_I^{\alpha}(y). \end{aligned}$$

The proof is by combining Theorem 1 of [8] with (2). In particular, for every $k < n - O(\log n)$ there are strings y of length n and complexity k such that $TCC_I^{\alpha}(y) > n - \alpha$ for all $\alpha < k - O(\log n)$ while $TCC_I^{\alpha}(y) = O(1)$ for $\alpha \ge k + O(1)$. We call such strings y *non-communicable*. For example, with $k = (\log n)^2$ this shows that there are y of complexity $C(y) = (\log n)^2$ with $TCC_I^{\alpha}(y) = n - (\log n)^2$ for all $\alpha < C(y) - O(\log n)$ and $O(1)$ otherwise. That is, Bob can hold a highly compressible string y, but cannot use that fact to reduce the communication complexity significantly below $|y|$! Unless *all* information about y is hard wired in the (total) protocol the communication between Bob and Alice requires sending y almost completely literally. For such y, irrespective of x, the communication complexity is *exponential* in the complexity of y for all protocols of complexity less that that of y; when the complexity of the protocol is allowed to pass the complexity of y then the communication complexity suddenly drops to 0.

Corollary 1. *For every n, k with $k \le n$ there are y of length n and $C(y) = k$ such that for every x $TCC_I^{\alpha}(x,y) \ge n - \alpha$ for $\alpha < C(y) - O(\log n)$; while for every x, y we have $TCC_I^{\alpha}(x,y) = O(1)$ for $\alpha \ge C(y) + O(1)$.*

This follows by combining Theorems 4, 5. If we relax the requirement of total and correct protocols to partial and partially correct protocols then we obtain the significantly weaker statements of Theorem 6 and Corollary 2.

5 Partially correct and partial protocols

The individual communication complexity can decrease if we do not require the communication protocol to be correct on all the input pairs. Let $CC_f^{\alpha}(x,y)$ stand for the minimum $CC^P(x,y)$ over all P of complexity at most α computing f correctly on input (x,y) (on other inputs P may output incorrect result). The minimum of the empty set is defined as ∞. Let $CC_{f,\text{1-way}}^{\alpha}(x,y)$ stand for the minimum $CC^P(x,y)$ over all one-way (Bob sends a message to Alice) P of complexity at most α computing $f(x,y)$ (again, on other inputs P may work incorrectly). For instance, if f is a Boolean function then $CC_{f,\text{1-way}}^{O(1)}(x,y) = 0$ for all x, y (either the protocol outputting always 0 or the protocol outputting always 1 is computes $f(x,y)$ for specific pair (x,y)).

5.1 Partially correct and partial protocols versus total ones

It is easy to see that in computing the Identity function for some (x,y) total, partially correct, protocols are more powerful than totally correct ones. A total partially correct protocol P computes $f(x,y)$ correctly for some (x,y), but may err on some inputs (u,v), in which case we set $CC^P(x,y) = \infty$. Being total such a protocol terminates for all inputs.

Definition 2. Let α be a natural number parameter. Let $CC_f^\alpha(x,y)$ stand for the minimum $CC^P(x,y)$ over all total partially correct protocols P of complexity at most α.

For instance, for every n there is a total protocol $P = P_n$ computable from n such that $CC^P_{I,\text{1-way}}(x,x) = 0$ (Alice outputs her string), thus $CC_I^{O(1)}(x,x) = 0$. On the other hand, for random x of length n we have $TCC_I^\alpha(x,x) \geq TCC_I^{\alpha-O(1)}(x) \geq C(x|n) - \alpha - O(\log\alpha) \geq n - \alpha - O(\log\alpha)$.

We also consider partial protocols that on some x,y are allowed to get stuck, that is, give no instructions at all about how to proceed. Formally, such a protocol is a pair of programs (P_A, P_B). The program P_A tells Alice what to do for each c (the current part of the conversation) and x: either wait the next bit from Bob, or to send a specific bit to Bob, or to output a certain string and halt. Similarly, the program P_B tells Bob what to do for each c and y: either to wait the next bit from Alice or to send a bit to Alice, or to halt. This pair must satisfy the following requirements for all $(x,y) \in \{0,1\}^n$ and all c: if a party gets the instruction to send a bit then another party gets the instruction to wait for a bit. However we do not require that for all $(x,y) \in \{0,1\}^n$ and all c both parties get some instruction, it is allowed that P_A, P_B start some endless computation. In particular, Alice may wait for a bit and at the same time Bob has no instruction at all.

Definition 3. The complexity of a partial protocol $P = (P_A, P_B)$ is defined as $C(P|n)$. We say that P computes f on the input (x,y) if Alice outputs $f(x,y)$ when P_A, P_B are run on (x,y). On other pairs Alice is allowed to output a wrong answer or not output anything at all. If protocol P does not terminate, or gives an incorrect answer, for input (x,y), then $CC^P(x,y) = \infty$. Two-way and one-way individual communication complexities with complexity of the partial protocol upper bounded by α are denoted as $PCC_f^\alpha(x,y)$ and $PCC^\alpha_{f,\text{1-way}}(x,y)$ respectively.

Since the total, partially correct, protocols are a subset of the partial protocols, we always have $PCC_f^\alpha(x,y) \leq CC_f^\alpha(x,y) \leq TCC_f^\alpha(x,y)$. Consider again the Identity function. We have the following obvious lower bound

$$C(y|x) - \alpha - O(\log\alpha) \leq PCC_I^\alpha(x,y) \tag{5}$$

for all α, x, y. On the other hand we have the following upper bound if α is at least $\log C(y) + O(1)$:

$$PCC^{\log C(y)+O(1)}_{I,\text{1-way}}(x,y) \leq C(y). \tag{6}$$

Indeed, we hardwire the value $C(y)$ in the protocol using $\log C(y)$ bits. This enables P_B to find a shortest description y^* of y and to send it to Alice; subsequently P_A decompresses the message received from Bob. Note that the program P_B gives no instruction to Bob if the complexity of Bob's input is greater than $C(y)$. Therefore, this protocol is not total. Comparing Equation (6) to Equation (3) we see that for PCC we have a better upper bound than for TCC. It turns out that for some pairs (x,y) the communication complexity for totally correct (and even for partially correct) protocols is close to the upper bound $n-\alpha$ while the communication complexity for partial protocols is close to the lower bound $C(y|x) \approx \alpha \ll n$.

Theorem 6. *For all α, n, x there are y of length n such that $CC_I^\alpha(x,y) \geq n - \alpha$ and $C(y|x) \leq \alpha + O(1)$.*

Proof. Fix a string x. By counting arguments, there is a string y with $CC_I^\alpha(x,y) \geq n - \alpha$. Indeed, there are less than $2^{\alpha+1}$ total protocols of complexity at most α. For each total protocol P there are at most $2^{n-\alpha-1}$ different y's with $CC^P(x,y) < n - \alpha$. Therefore the total number of y's with $CC_I^\alpha(x,y) < n-\alpha$ is less than $2^{\alpha+1}2^{n-\alpha-1} = 2^n$.

Let y be the first string with $CC_I^\alpha(x,y) \geq n - \alpha$. To identify y conditional to x we only need to now the number of total protocols of complexity at most α: given that number we enumerate all such protocols until we find all them. Given all those protocols and x we run all of them on all pairs (x,y) to find $CC_I^\alpha(x,y)$ (here we use that the protocols are total) for every y, and determine the first y for which it is at least $n-\alpha$. Hence $C(y|x) \leq \alpha + O(1)$. ☐

Corollary 2. *Fix constants C_1, C_2 such that $CC^{\log C(y)+C_1}_{I,\text{1-way}}(x,y) \leq C(y) \leq n + C_2$. Applying the theorem to the empty string ϵ and to (say) $\alpha = 2\log n$ we obtain a y of length n with exponential gap between $CC_I^{2\log n}(\epsilon,y) \geq n - 2\log n - O(1)$ and $PCC^{\log(n+C_2)+C_1}_{I,\text{1-way}}(\epsilon,y) \leq C(y) \leq \log n + O(1)$.*

Using a deep result of An. Muchnik [7] we can prove that $PCC^\alpha_{I,\text{1-way}}$ is close to $C(y|x)$ for $\alpha \geq O(\log n)$.

Theorem 7 (An. Muchnik). *For all x, y of length n there is p such that $|p| \leq C(y|x) + O(\log n)$, $C(p|y) = O(\log n)$ and $C(y|p, x) = O(\log n)$, where the constants in $O(\log n)$ do not depend on n, x, y.*

Corollary 3. *For all x, y of length n we have $PCC_{I,1\text{-}way}^{O(\log n)}(x, y) \leq C(y|x) + O(\log n)$.*

Proof. Let p be the program of Muchnik's theorem, let q be the program of length $O(\log n)$ for the reference computer to reconstruct p from y and let r the program of length $O(\log n)$ for the reference computer to reconstruct y from the pair (x, p). The protocol is as follows: Bob finds p from y, q and sends p to Alice; Alice reconstructs y from x, r. Both q and r are hardwired into the protocol, so its complexity is $O(\log n)$. This protocol is partial, as both Bob and Alice may be stuck when reconstructing p from y', q and y from x', r. □

For very small values of $C(y|x), C(y)$ we can do even better using the coloring lemma 3.9 and theorem 3.11 from [1].

Lemma 1. *Let k_1, k_2 be such that $C(x) \leq k_1$ and $C(y \mid x) \leq k_2$, and let $m = |\{(x, y) : C(x) \leq k_1,\ C(y \mid x) \leq k_2\}|$. For $M = 2^{k_1}$, $N = 2^{k_2}$ and every $1 \leq B \leq N$ Bob can compute the recursive function $R(k_1, k_2, m, y) = r_y \leq (N/B)e(MN)^{1/B}$ such that Alice can reconstruct y from x, r_y, m and at most $b \leq \log B$ extra bits.*

Using k_1, k_2, m, y, Bob can compute r_y and send it in $\log r_y$ bits to Alice. The latter computes y from x, m, r_y using additionally $b \leq \log B$ special bits provided by the protocol. Then, the number of bits that need to be communicated, 1 round, from Bob to Alice, is

$$\log r_y \leq k_2 - \log B + \frac{k_2 + k_1}{B}.$$

The protocol $P = (P_A, P_B)$ uses $\leq 2(k_1 + k_2) + b + O(1)$ bits.

Corollary 4. *If $C(x), C(y|x) = O(\log \log n)$ and $b = \Theta(\log \log n)$ then $PCC_{I,1-way}^{\Theta(\log \log n)}(x, y) \leq C(y|x) - \Theta(\log \log n)$.*

5.2 Two-way is better than one-way for partially correct protocols

Note that for the Identity function all our upper bounds hold for one-way protocols and all our lower bounds hold for two-way protocols. The following question arises: are two-way protocols more powerful than one-way ones (to compute the Identity function)? Theorem 4 implies that for total protocol it does not matter whether the communication is one-way or two-way. For partially correct total protocols and partial protocol the situation is different. It turns out that partially correct total two-way protocols are stronger than even partial one-way protocols.

Theorem 8. *For every k, l, s such that $k \geq s + l2^s$ there are strings x, y of length $(2^s + 1)k$ such that $CC_I^{O(1)}(x, y) \leq 2^s \log(2k)$ but $PCC_{I,1\text{-}way}^{s}(x, y) \geq l$.*

Proof. We let $x = z_0 z_1 \dots z_{2^s}$ where $z_0, \dots, z_{2^s}$ have length k and $y = z_j 00 \dots 0$ for some j.

To prove the upper bound consider the following two-way protocol: Alice finds a set of indexes $I = \{i_1, \dots, i_{2^s}\}$ such that for every distinct j, m there is $i \in I$ such that ith bit of z_j is different from ith bit of z_m (such set does exist, which may be shown by induction). Then she sends to Bob the string $i_1 \dots i_{2^s}$ and Bob sends to Alice ith bit of y for all $i \in I$. Alice knows now y.

We need to find now particular $z_0, z_1, \dots, z_{2^{a+b+s}}$ such that no one-way protocol is effective on the pair (x, y) obtained from them in the specified way. To this end let $P_1, \dots, P_N$ be all the one-way partial protocols of complexity less than s computing the identity function. For every z and $i \leq N$ let $c(z, i)$ denote the message sent by Bob in protocol P_i when he receives $z00\dots0$ as help bits provided the length of the message is less than l. Otherwise let $c(z, i) = \infty$. Let $c(z)$ stand for the concatenation of $c(z, i)$ over all i. The range of $c(z)$ has $(2^l)^N < 2^{l2^s}$ elements. Hence there is c such that for at least $2^{k-2^s l} > 2^s$ different z's we have $c(z) = c$. Pick such c and pick different $z_0, z_1, \dots, z_{2^s}$ among those z's. Let y_j stand for the string obtained from z_j by appending 0s. We claim that $CC_I^{P_i}(x, y_j) \geq l$ for some j for all $i \leq N$. Assume that this is not the case. That is, for every j there are i such that $CC_I^{P_i}(x, y_j) < l$. There are $j_1 \neq j_2$ for which i is the same. As $c(z_{j_1}, i) = c(z_{j_2}, i) \neq \infty$ Alice receives the same message in P_i on inputs (x, y_{j_1}), (x, y_{j_2}) and should output both answers y_{j_1}, y_{j_2}, which is a contradiction. □

Corollary 5. *Let in the above theorem $s=(\log k)/3$ and $l=k^{2/3}/\log k$. These values satisfy the condition $k\ge s+l2^s$ and hence there are x,y of length about $k^{4/3}$ with almost quadratic gap between $CC_I^{O(1)}(x,y)\le k^{1/3}\log k$ and $PCC_{I,\textit{1-way}}^{(\log k)/3}(x,y)\ge k^{2/3}/\log k$. Letting $s=\log\log k$ and $l=k/(2\log k)$ we obtain x,y of length about $k\log k$ with an exponential gap between $CC_I^{O(1)}(x,y)\le \log k\log(2k)$ and $PCC_{I,\textit{1-way}}^{\log k}(x,y)\ge k/(2\log k)$.*

6 Summary of some selected results for comparison

- $\forall_{x,y,\alpha}[TCC_I^\alpha(x,y)\ge CC_I^\alpha(x,y)\ge PCC_I^\alpha(x,y)]$ by definition.
- $\forall_{\alpha,x,y}[TCC_I^{\alpha+O(1)}(y)=TCC_I^\alpha(x,y)+O(1)]$ Theorem 4 and discussion.
- $\forall_{n,k,\alpha}\exists_{y,|y|=n,C(y)=k}\forall_x[\alpha<C(y)-O(\log n)\Rightarrow TCC_I^\alpha(x,y)\ge n-\alpha]$ Corollary 1.
- $\forall_{x,y,\alpha}[\alpha\ge C(y)-O(1)\Rightarrow TCC_I^\alpha(x,y)=O(1)]$ Corollary 1.
- $\forall_{n,x,\alpha}\exists_{y,|y|=n}[\alpha\ge C(y|x)-O(1)\& CC_I^\alpha(x,y)\ge n-\alpha]$ Theorem 6.
- $\forall_{x,y,\alpha}[PCC_I^\alpha(x,y)\ge C(y|x)-\alpha-O(\log\alpha)]$ (5).
- $\forall_{n,x,y}[PCC_{I,1-way}^{\log C(y)+O(1)}(x,y)\le C(y)]$ (6).
- $\forall_n\forall_{x,y,|x|=|y|=n}[PCC_{I,1-way}^{O(\log n)}(x,y)\le C(y|x)+O(\log n)]$ Corollary 3.
- $\forall_{k,l,s:k\ge s+l2^s}\exists_{x,y,|x|=|y|=(2^s+1)k}[CC_I^{O(1)}(x,y)\le 2^s\log(2k)\& PCC_{I,1-way}^s(x,y)\ge l]$ Theorem 8.

7 Protocols using help bits

A (partial) protocol with a help bits for Alice b help bits for Bob on strings of length n may be defined as a regular (partial) protocol P on inputs u,v of length $n+a,n+b$, respectively. We say that P computes f on the input (x,y) if there are $h_A\in\{0,1\}^a$ and $h_B\in\{0,1\}^b$ such according to P on input (xh_A,yh_B) Alice outputs $f(x,y)$. The crucial point in this definition is that the help bit sequences h_A,h_B may depend on the input pair (x,y). We say that P computes f if it computes f on all input pairs $(x,y)\in\{0,1\}^n$. Thus P may compute many different functions. For instance there is a protocol with 1 help bit for Alice and no help bits for Bob that computes every Boolean function: Alice just receives the value of the function as the help bit and outputs it. Define $CC^P(x,y)$ as the minimum of the length of conversation on input (xh_A,yh_B) according to P over $h_A\in\{0,1\}^a$, $h_B\in\{0,1\}^b$ such that on input (xh_A,yh_B) Alice outputs $f(x,y)$ (the minimum of the empty set is defined as ∞). We define $TCC_f^{\alpha,a,b}(x,y)$ as the minimum $CC^P(x,y)$ over all P of complexity at most α computing f (over all inputs, and not only on (x,y)). Define $TCC_{f,\text{1-way}}^{\alpha,a,b}(x,y)$ $CC_f^{\alpha,a,b}(x,y)$, $CC_{f,\text{1-way}}^{\alpha,a,b}(x,y)$ analogously (in the latter two we minimize over all P of complexity at most α).

7.1 Partially correct protocols versus total ones—with help bits

In contrast to the no-help-bit case, now the difference between totally and partially correct protocols in not essential: allowing only one extra help bit we can effectively transform a protocol P computing f on specific input (x,y) into a protocol P' computing f an all inputs so that $TCC_f^{P'}(x,y)\le CC_f^P(x,y)$: at first Alice and Bob receive 1 bit of help information (or only one of them, and in that case he/she resends that bit to the other; in this case the right hand side of the inequality should be incremented by 1). If this is the case then they start P. Otherwise they start the default protocol. So we obtain

$$\begin{aligned}TCC_f^{\alpha+O(1),a+1,b+1}(x,y)&\le CC_f^{\alpha,a,b}(x,y),\\ TCC_f^{\alpha+O(1),a,b+1}(x,y)&\le CC_f^{\alpha,a,b}(x,y)+1,\\ TCC_f^{\alpha+O(1),a+1,b}(x,y)&\le CC_f^{\alpha,a,b}(x,y)+1.\end{aligned}$$

The same applies to $TCC_{\text{1-way}}$ and $CC_{\text{1-way}}$ (except the last inequality, as now Alice is unable to send to Bob). Therefore we will not consider specially totally correct protocols. We will study only values $CC_f^{\alpha,a,b}(x,y)$, $CC_{f,\text{1-way}}^{\alpha,a,b}(x,y)$.

Moreover, we can decrease a by a' at the expense of increasing α by $a'+b'+O(\log b')$ (help bits are appending to the program specifying the protocol), and similarly for b:

$$CC_f^{\alpha+a'+b'+O(\log a'b'),a,b}(x,y) \le CC_f^{\alpha,a+a',b+b'}(x,y),$$

but not vice verse. The same is true for 1-way protocols.

7.2 Partial protocols with help bits

For partial protocol we can even decrease α at the expense of increasing both a and b: indeed let p be the shortest program for (P_A, P_B) and let q we the prefix of p and r be the remaining bits of p. Consider now the following programs P'_A, P'_B; both receive r as help an both have p hard wired. P_A appends computes $p=qr$ and decompresses p and then executes P_A. The program P_B acts in a similar way. Note that $C(P'_A, P'_B) \le |q| + O(1)$. Thus we obtain

$$PCC_f^{\alpha+O(1),a+\alpha'+O(\log\alpha'),b+\alpha'+O(\log\alpha')}(x,y) \le PCC_f^{\alpha+\alpha',a,b}(x,y).$$

7.3 Two-way is better than one-way with help bits

Theorem 9. *For every k,l,s,a,b such that $k \ge a+b+s+l2^{s+b}$ there are strings x,y of length $(2^{a+b+s}+1)k$ such that there is a two-way protocol of complexity $O(1)$ with 1 help bit (either for Alice or for Bob) such that $CC^P(x,y) \le 2^{a+b+s}\log(2k)+1$ but for every one-way protocol P of complexity less than s with a help bits for Alice and b help bits for Bob we have $CC^P(x,y) \ge l$.*

Proof. The proof is similar to the proof of the previous Theorem 8.

We let $x = z_0 z_1 \dots z_{2^{a+b+s}}$ where $z_0, \dots, z_{2^{a+b+s}}$ have length k and $y = z_j 00\dots 0$ for some j.

To prove the upper bound consider the following two-way protocol: if x,y has not the above form Alice receives 0 as the help bit and starts the default protocol. Otherwise she receives 1 as the help bit and finds a set of indexes $I = \{i_1, \dots, i_{2^{a+b+s}}\}$ such that for every distinct j,m there is $i \in I$ such that ith bit of z_j is different from ith bit of z_m. Then she sends to Bob the string $1i_1 \dots i_{2^{a+b+s}}$ and Bob sends to Alice ith bit of y for all $i \in I$. Alice knows now y.

We need to find now particular $z_0, z_1, \dots, z_{2^{a+b+s}}$ such that no one-way protocol is effective on the pair (x,y) obtained from them in the specified way. To this end let $P_1, \dots, P_N$ be all the one-way protocols of complexity less than s with a help bits for Alice and b help bits for Bob computing the identity function. For every z, $i \le N$ and h_B where h_B is a binary sequence of length b let $c(z,i,h_B)$ denote the message sent by Bob in protocol P_i when he receives $z00\dots0$ as the input and h_B as help bits provided the length of the message is less than l. Otherwise let $c(z,i,h_B) = \infty$. Let $c(z)$ stand for the concatenation of $c(z,i,h_B)$ over all i, h_B. The range of $c(z)$ has $(2^l)^{N2^b} < 2^{l2^{s+b}}$ elements. Hence there is c such that for at least $2^{k-2^{s+b}l} > 2^{a+b+s}$ different z's we have $c(z) = c$. Pick such c and pick different $z_0, z_1, \dots, z_{2^{a+b+s}}$ among those z's. Let y_j stand for the string obtained from z_j by appending 0s. We claim that $TCC^{P_i}(x,y_j) \ge l$ for some j for all $i \le N$. Assume that this is not the case. That is, for every j there are i, h_A, h_B such that $TCC^{P_i}(x,y_j) < l$ with help bit sequences h_A, h_B. There are $j_1 \ne j_2$ for which the triples (i, h_A, h_B) coincide. As $c(z_{j_1}, i, h_B) = c(z_{j_2}, i, h_B) \ne \infty$ Alice receives the same message in P_i on inputs (x, y_{j_1}), (x, y_{j_2}), with the help bit sequences h_A, h_B and should output both answers y_{j_1}, y_{j_2}, which is a contradiction. $\square$

Corollary 6. *Let in the above theorem $a=b=s=(\log k)/6$, $l = k^{1/2}/\log k$. These values satisfy the condition $k \ge a+b+s+l2^{s+b}$ and hence there are x,y of length about $k^{1.5}$ for which there is a two-way protocol of complexity $O(1)$ with only one help bit with $CC^P(x,y) \le k^{1/3}\log k$ but there is no one-way protocol of complexity $(\log k)/6$ with $(\log k)/6$ help bits both for Alice and Bob with $CC^P(x,y) < k^{1/2}/\log k$.*

References

[1] C.H. Bennett, P. Gács, M. Li, P.M.B. Vitányi, and W. Zurek, Information Distance, *IEEE Trans. Information Theory*, IT-44:4(1998) 1407–1423.

[2] H. Buhrman, T. Jiang, M. Li, P. Vitanyi, New applications of the incompressibility method: Part II, *Theoretical Computer Science*, 235:1(2000), 59–70.

[3] P. Gács, J. Tromp, P.M.B. Vitányi. Algorithmic statistics, *IEEE Trans. Inform. Th.*, 47:6(2001), 2443–2463.

[4] E. Kushilevitz, N. Nisan, *Communication Complexity*, Cambridge University Press, 1997.

[5] L. A. Levin, Average Case Complete Problems, *SIAM J.Comput.*, 15:1(1986), 285–286.

[6] M. Li and P.M.B. Vitányi, *An Introduction to Kolmogorov Complexity and its Applications*, Springer-Verlag, New York, 2nd Edition, 1997.

[7] An.A. Muchnik, Conditional complexity and codes, *Theoret. Comput. Sci.*, 271:1/2(2002), 97–111.

[8] N.K. Vereshchagin and P.M.B. Vitányi, Kolmogorov's structure functions and an application to the foundations of model selection, *Proc. 47th IEEE Symp. Found. Comput. Sci.*, 2002, 751–760.

[9] A.C. Yao. Some complexity questions related to distributive computing. In: *Proc. 11th ACM Symposium on Theory of Computing*, 1979, 209–213.