On the P versus NP intersected with co-NP question in communication complexity

Stasys Jukna

Abstract

We consider the analog of the P versus NP $\cap$ co-NP question for the classical two-party communication protocols where polynomial time is replaced by poly-logarithmic communication: if both a boolean function f and its negation $\neg f$ have small (poly-logarithmic in the number of variables) nondeterministic communication complexity, what is then its deterministic and/or probabilistic communication complexity? In the *fixed* (worst) partition model of communication this question was answered by Aho, Ullman and Yannakakis in 1983: here P = NP $\cap$ co-NP.

We show that in the *best* partition model of communication the situation is entirely different: here P is a *proper* subset even of RP $\cap$ co-RP. This, in particular, resolves an open question raised by Papadimitriou and Sipser in 1982.

1 Introduction

Understanding the relative power of determinism, nondeterminism, and randomization is fundamental in any model of computation. In the Turing machine model this leads to the well-known P versus NP versus BPP and similar questions. While in this model such questions remain widely open, some progress was made in several much simpler (but still important) models, like decision trees, restricted branching programs or communication protocols.

In the decision tree model when the complexity measure is the *depth* of a tree we have that P = NP $\cap$ co-NP [3, 5, 19]: if both f and $\neg f$ can be computed nondeterministic decision trees of depth at most d then f can be computed by a deterministic

[1] *Current address*: Universität Frankfurt, Institut für Informatik Robert-Mayer-Str. 11-15, D-60054 Frankfurt, Germany. Email: jukna@thi.informatik.uni-frankfurt.de

[2] Institute of Mathematics and Informatics, Akademijos 4, LT-08663 Vilnius, Lithuania.

[3] Research supported by a DFG grant SCHN 503/2-2.

decision tree of depth at most d^2. Nisan [14] has shown that in this case also P = BPP holds: if f can be computed by a probabilistic bounded error decision tree of depth d then f can be computed by a deterministic decision tree of depth $O(d^3)$.

Interestingly, the situation is different if we measure the *size* of (the total number of vertices in) a tree instead of its depth—then $\mathrm{P} \neq \mathrm{NP} \cap \text{co-NP}$ [8]: there are explicit boolean functions f such that both f and $\neg f$ can be computed by nondeterministic decision trees of size at most N but any deterministic decision tree for f has size $N^{\Omega(\log N)}$. A similar situation is when the complexity measure of a decision tree is the number of its non-isomorphic subtrees (this model corresponds to so called read-once branching programs—just merge isomorphic subtrees): here we also have that $\mathrm{P} \neq \mathrm{NP} \cap \text{co-NP}$ [8], $\mathrm{BPP} \not\subseteq \mathrm{NP}$ [17] and $\mathrm{NP} \cap \text{co-NP} \not\subseteq \mathrm{BPP}$ [18].

In this note we consider the classical model of two-party communication protocols introduced by Yao [21] (see [10] for more information). There are two main types of such protocols: the **fixed partition** type where the protocol must use some prescribed (by an adversary) "bad" partition of input variables between the players, and **best partition** type where the protocol is allowed to choose the "most suitable" for a given function partition of its variables.

Although historically the best partition model of communication has reveived less attention than the fixed partition model, the former one has larger applicability. This model naturally arises when dealing with time-space trade-offs of VLSI chips (see, e.g., [12]). It (also naturally) arises in the context of branching programs. In fact, most of lower bounds for various restricted models of branching programs were obtained by proving (more or less explicitely) the corresponding lower bounds on the communication complexity of different types of best partition protocols (see [20] for a comprehensive description of such applications).

As in other models of computation, there are three natural modes of communication: deterministic, nondeterministic and probabilistic. Having these modes and having the (admittedly far-fetched) analogy with the P versus NP question, it is natural to consider the relations between the corresponding complexity classes. Here for convenience (and added thrill, just like in [2]) we use the common names for the analogs of the complexity classes:

> Let P, NP, RP, and BPP consist, respectively, of all boolean functions in n variables whose deterministic, nondeterministic, probabilistic one-sided error and probabilistic bounded error communication complexity is polynomial in $\log n$.

In the **fixed partition** case most of these problems are already solved (of course, this has nothing to do with the relations between Turing machine classes). The separation $\mathrm{NP} \neq \text{co-NP}$, and hence, $\mathrm{P} \neq \mathrm{NP}$ can be easily shown using the equality function $\textsc{eq}(x, y)$ which tests whether two given binary strings x and y of length n

are equal. A less obvious separations $\mathrm{P} \neq \mathrm{RP}$ and $\mathrm{BPP} \not\subseteq \mathrm{NP}$ was shown by Rabin and Yao (see [22]): $\neg\textsc{eq}(x,y)$ can be computed by a probabilistic (even one-sided $1/n$-error) protocol with only $O(\log n)$ bits of communication. The incomparability of NP and BPP was shown by Babai, Frankl, and Simon [2] using the set-disjointness function $\textsc{disj}(x,y)$, which outputs 1 iff $\sum_{i=1}^{n} x_i y_i = 0$. They proved that this function has probabilistic bounded error communication complexity $\Omega(\sqrt{n})$. This was later improved to $\Omega(n)$ by Kalyanasundaram and Schnitger [9]; a simpler proof was found by Razborov [16].

Interestingly, the analog of the P versus $\mathrm{NP} \cap \text{co-NP}$ question in the fixed partition case has a *positive* answer. This is a direct consequence of the following well-known result of Aho, Ullman, and Yannakakis [1]: if f and $\neg f$ have nondeterministic communication complexities n_f and $n_{\neg f}$, then the deterministic communication complexity of f does not exceed $O\left(\max\{n_f, n_{\neg f}\}^2\right)$. Hence, in the fixed partition case we have

$$\mathrm{P} = \mathrm{ZPP} = \mathrm{RP} \cap \text{co-RP} = \mathrm{NP} \cap \text{co-NP} \subsetneqq \mathrm{BPP}. \tag{1}$$

The **best partition** model is more difficult to analyze, and here the situation was less clear. In particular, such "clean" functions like $\textsc{eq}(x,y)$ or $\textsc{disj}(x,y)$ cannot be used (at least directly) for separations anymore just because even the deterministic communication complexity of these functions is constant. On the other hand (as we already mentioned above), this model is important because it is closely related to the area/time complexity of VLSI circuits and to the width of branching programs. Thus, it is worthwhile to investigate the relationship among complexity classes in the best partition model.

The first separation in this model was given by Papadimitriou and Sipser in [15] by proving an earlier conjecture of Lipton and Sedgewick [13] that the triangle-freeness property of graphs is hard for nondeterministic best partition protocols. This first showed that $\mathrm{NP} \neq \text{co-NP}$, and hence, also $\mathrm{P} \neq \mathrm{NP}$ in this model. The proof was via a reduction to computing $\textsc{disj}(x,y)$ in the fixed-partition case. The result was further extended in [4, 7] to the case where a protocol is allowed to use different partitions for different inputs.

Using shifted versions of $\textsc{eq}(x,y)$ and $\textsc{disj}(x,y)$, Ja'Ja', Prasanna Kumar and Simon [6] showed that $\mathrm{P} \subsetneqq \mathrm{RP} \subsetneqq \mathrm{NP}$ holds in the best partition case. Using similar techniques, Lam and Ruzzo [11] have proved that NP and BPP remain incomparable also in the best partition case.

In the same paper [15], Papadimitriou and Sipser asked whether $\mathrm{P} \neq \mathrm{NP} \cap \text{co-NP}$ for the best partition protocols. As noted in [1], the question is important because it exposes something about the power of lower bound arguments. We can prove a lower bound on the deterministic communication complexity of a function f by arguing about either f or $\neg f$. But if both the function and its negation have low

nondeterministic complexity under *some* partitions of variables, other arguments are needed to show that the deterministic communication complexity must be large for *any* partition.

That an appropriate modification of the triangle-freeness function could separate P from NP $\cap$ co-NP in the best partition case was claimed in [1]. Unfortunately, the proof—which should (apparently) involve the argument of [15] for the triangle-freeness function—was never given, and the question remained open (cf., e.g., [11]).

2 Our results

In this note we prove that P $\neq$ NP $\cap$ co-NP for best partition protocols. Actually, we establish even stronger separations showing that in the best partition case the situation is entirely different: the possibility to use different partitions for f and $\neg f$ *can* exponentially increase the power of randomness as well as of nondeterminism. Namely, in the best partition case we have the following separations (cf. (1)):

$$\mathrm{P} \subsetneqq \mathrm{RP} \cap \text{co-RP} \subsetneqq \mathrm{NP} \cap \text{co-NP} \not\subseteq \mathrm{BPP}. \tag{2}$$

All this is a direct consequence of the following theorem. We adopt the following convention for discussing different communication complexity measures of f in the *best partition* case: $D(f)$ for the deterministic, $N(f)$ for the nondeterministic, $R(f)$ for the probabilistic bounded error, and $R^1(f)$ for probabilistic one-sided error communication complexity.

Theorem 2.1. *There are explicit boolean functions f and g in n^2 variables such that:*

(i) *both $R^1(f)$ and $R^1(\neg f)$ are $O(\log n)$ but $D(f) = \Omega(n)$;*

(ii) *both $N(g)$ and $N(\neg g)$ are $O(\log n)$ but $R(g) = \Omega(n)$.*

Moreover, the upper bounds hold for one-round protocols.

The rest is devoted to the proof of this theorem. We do this by reductions to known upper and lower bounds in the *fixed* partition models of functions such as Equality $\textsc{eq}(x, y)$ and Set Disjointness $\textsc{disj}(x, y)$. The proofs themselves are quite simple—as it often happens with the results of this type, most of the work is done by a careful choice of a "right" separating function.

Recall that in the best partition case the players can choose different (most suitable) partitions for a function and its negation. To visualize the effect of this choice, in both cases we define the corresponding separating function $f(X)$ as boolean functions in n^2 variables, arranged into an $n \times n$ matrix (we assume that n is sufficiently

large). Hence, inputs for f are 0/1 matrices $A : X \to \{0,1\}$. We define $f(X)$ in such a way that a partition of X according to columns is suitable for computing f, and that according to rows is suitable for $\neg f$.

3 Proof of Theorem 2.1(i)

We define the boolean function $f(X)$, showing that P $\subsetneq$ RP $\cap$ co-RP in the best partition case, as follows. Inputs for f are $n \times n$ matrices $A : X \to \{0,1\}$; this time we require that n is even. Say that a row/column of such a matrix is *odd* (*even*) if it contains an odd (even) number of 1's. Let $f(A) = 1$ if and only if A has at least one odd row, and all columns of A are odd.

Lemma 3.1. *Both $R^1(f)$ and $R^1(\neg f)$ are $O(\log n)$.*

Proof. In the protocol for f Alice takes the first half of *columns* whereas in the protocol for $\neg f$ she takes the first half of *rows*. After that the computation of f and $\neg f$ reduces to the computation of the non-equality function $\neg\text{EQ}(x,y)$, where x is a string of parities of rows/columns seen by Alice and y is a string of parities of rows/columns seen by Bob. Indeed, to compute f it is enough to decide whether $x \neq y$ (there is an odd row) whereas for $\neg f$ it is enough to decide whether $x \neq y \oplus \mathbf{1}$ (there is an even column). This completes the proof because, as we already mentioned above, $\neg\text{EQ}(x,y)$ can be computed with a probabilistic one-sided error protocol by communicating only $O(\log n)$ bits. □

Lemma 3.2. $D(f) = \Omega(n)$.

Proof. Take an arbitrary deterministic protocol for $f(X)$. The protocol uses some balanced partition of the entries of X into two halves where the first half is seen by Alice and the second by Bob. Say that a column is seen by Alice (resp., Bob) if Alice (resp., Bob) can see all its entries. A column is *mixed* if it is seen by none of the two players, that is, if each player can see at least one of its entries. Let m be the number of mixed columns. We consider two cases depending on how large this number m is. In both cases we describe a "hard" subset of inputs, i.e. a subset of input matrices on which the players need to communicate many bits.
Case 1: $m \leq n/2 - 1$. Since each player can see at most $n/2$ columns, we have that in this case each player will see at least $n - (n/2 + m) \geq 1$ columns. Take one column seen by Alice and another column seen by Bob, and let Y be the $(n-1) \times 2$ submatrix of X formed by these two columns without the last row r. We restrict the protocol to input matrices $A : X \to \{0,1\}$ defined as follows. We set to 1 all entries in the last row r, and set to 0 all remaining entries of X outside Y. The columns x and y of Y may take arbitrary values such that the resulting vectors are even. This way

we ensure that all columns of A are odd. Moreover, the last row r is even since n is even. Thus, given such a matrix A, the players must determine whether some of the remaining rows is odd. That is, they must determine whether $x \neq y$, which requires $\Omega(n)$ bits of communication.
Case 2: $m \geq n/2$. Let M be the $n \times m$ submatrix of X formed by the mixed columns. Select from the i-th ($i = 1, \ldots, m$) column of M one entry x_i seen by Alice and one entry y_i seen by Bob. Since $m \leq n$ and we select only $2m$ entries, there must be a row r with $t \leq 2$ selected entries. Let Y be the $n \times (m-t)$ submatrix of M consisting of the mixed columns with no selected entries in this row r. We may assume that $m - t$ is odd (if not, then just include one column less in Y).

Now restrict the protocol to input matrices $A : X \to \{0,1\}$ defined as follows. First we set the part of the row r lying in Y to 0's and the rest of r to 1's. Since n is even and $m - t$ is odd, this ensures that the obtained matrices will already contain an odd row. After that we set to 0 all the remaining non-selected entries of X. Since each obtained matrix A contains an odd row (the row r) and all columns outside the submatrix Y are odd (each of them has a 1 in the row r and 0's elsewhere), the players must determine whether all columns of A in Y are also odd. That is, they must determine whether $x_i \neq y_i$ for all $i = 1, \ldots, m - t$. Or equivalently, they must decide whether $x = y \oplus \mathbf{1}$ for vectors $x = (x_1, \ldots, x_{m-t})$ and $y = (y_1, \ldots, y_{m-t})$, which again requires $\Omega(m - t) = \Omega(n)$ bits of communication.

This completes the proof of Lemma 3.2, and thus, the proof of the first claim of Theorem 2.1. □

4 Proof of Theorem 2.1(ii)

We define the boolean function $g(X)$, showing that NP ∩ co-NP $\not\subseteq$ BPP, and hence, RP ∩ co-RP $\subsetneq$ NP ∩ co-NP in the best partition case, as follows. Say that a row/column x of a 0/1 matrix is *good* if it contains precisely two 1's, and *bad* otherwise. Let $g(A) = 1$ if and only if at least one row of A is good, and all columns of A are bad.

Lemma 4.1. *Both $N(g)$ and $N(\neg g)$ are $O(\log n)$.*

Proof. To compute g and $\neg g$ the players use the same partitions of the input matrix as in the proof of Lemma 3.1.

To compute $g(A)$ for a given matrix $A : X \to \{0,1\}$, the protocol first guesses a row r (a candidate for a good row). Then, using 3 bits, Alice tells Bob whether all her columns are bad, and whether the first half of the row r contains none, one, two or more than two 1's. After that Bob has the whole information about the value $g(A)$, and can announce the answer. The negation $\neg g(A)$ can be computed in the same manner by replacing the roles of rows and columns. □

Lemma 4.2. $R(g) = \Omega(n)$.

Proof. The proof is almost the same as that of Lemma 3.2—the only difference is that now we use the set disjointness function $\text{DISJ}(x, y)$ instead of $\text{EQ}(x, y)$.

Take an arbitrary probabilistic bounded error protocol for $g(X)$ using some balanced partition of the $n \times n$ matrix of variables X. Let (as before) m be the number of mixed columns.

Case 1: $m \leq n/2 - 1$. In this case each player can see at least one column. Take one column seen by Alice and another column seen by Bob, and let Y be the $(n-3) \times 2$ submatrix of X formed by these two columns without the last three rows. We restrict the protocol to input matrices $A : X \to \{0, 1\}$ defined as follows. We first set all entries in the last three rows to 1. This way we ensure that all columns of A are already bad. Then we set all remaining entries of X outside Y to 0. The columns x and y of Y may take arbitrary values.

In each such matrix all columns are bad, and the last three all-1 rows are also bad (for $n \geq 3$). Thus, given such a matrix, the players must determine whether some of the remaining rows is good. Since all these rows have 0's outside the columns x and y, this means that the players must determine whether $x_i = y_i = 1$ for some $1 \leq i < n - 3$. That is, they must compute $\neg\text{DISJ}(x, y)$ which (according to [9, 16]) requires $\Omega(n)$ bits of communication.

Case 2: $m \geq n/2$. Let M be the $n \times m$ submatrix of X formed by the mixed columns. Select from the i-th ($i = 1, \ldots, m$) column of M one entry x_i seen by Alice and one entry y_i seen by Bob. As before, there must be a row r with $t \leq 2$ selected entries. Let Y be the $n \times (m - t)$ submatrix of M consisting of the mixed columns with no selected entries in the row r. We may assume that $m - t \leq n - 2$ (if not, then just include in Y fewer columns).

Now restrict the protocol to input matrices $A : X \to \{0, 1\}$ defined as follows. First we set to 1 some two entries of the row r lying outside Y, and set to 0 all the remaining entries of r. This ensures that the obtained matrices will already contain a good row. After that we set all the remaining non-selected entries of X to 0. Since each obtained matrix A contains a good row (such is the row r) and all columns outside the submatrix Y are bad (each of them can have a 1 only in the row r), the players must determine whether all columns of A in Y are also bad. Since all non-selected entries of Y are set to 0, the players must determine whether $x_i + y_i \leq 1$ for all $i = 1, \ldots, m-t$. Hence, the players must decide whether $\sum_{i=1}^{m-t} x_i y_i = 0$, that is, to compute the set-disjointness function $\text{DISJ}(x, y)$, which again requires $\Omega(m-t) = \Omega(n)$ bits of communication.

This completes the proof of Lemma 4.2, and thus, the proof of the second claim of Theorem 2.1. □

Acknowledgments

I would like to thank Hartmut Klauck and Martin Sauerhoff for helpful comments on an early version of this paper, and Georg Schnitger for motivating discussions. I am also thankful to anonymous referees for useful suggestions.

References

[1] Aho, A., Ullman, J. and Yannakakis, M. (1983): On notions of information transfer in VLSI circuits. In *Proc. of 15th Ann. ACM Symp. on the Theory of Computing*, 133–139.

[2] Babai, L., Frankl, P. and Simon, J. (1986): Complexity classes in communication complexity theory. In *Proc. of 27th Ann. IEEE Symp. on Foundations of Comput. Sci.*, 337–347.

[3] Blum, M. and Impagliazzo, R. (1987): Generic oracles and oracle classes. In *Proc. of 28th Ann. IEEE Symp. on Foundations of Comput. Sci.*, 118–126.

[4] Ďuriš, P., Hromkovič, J., Jukna, S., Sauerhoff, M. and Schnitger, G. (2004): On multipartition communication complexity, *Information and Computation* **194**:1, 49–75.

[5] Hartmanis, J. and Hemachandra, L. A. (1987): One-way functions, robustness and non-isomorphism of NP-complete classes. Tech. Rep. DCS TR86-796, Cornell University.

[6] Ja'Ja,' J., Prasanna Kummar, V. K. and Simon, J. (1984): Information transfer under different sets of protocols, *SIAM J. Comput.* **13**, 840–849.

[7] Jukna, S. and Schnitger, G. (2002): Triangle-freeness is hard to detect, *Combinatorics, Probability and Computing* **11**, 549–569.

[8] Jukna, S., Razborov, A., Savický, P. and Wegener, I. (1999): On P versus NP∩co-NP for decision trees and read-once branching programs, *Computational Complexity* **8**:4, 357–370.

[9] Kalyanasundaram, B. and Schnitger, G. (1992): The probabilistic communication complexity of set intersection, *SIAM J. Discrete Math.* **5**:4 (1992), 545–557.

[10] Kushilevitz, E. and Nisan, N. (1997): *Communication Complexity*. Cambridge University Press.

[11] Lam, T. W. and Ruzzo, W. L. (1992): Results on communication complexity classes, *J. Comp. Syst. Sci.* **44**, 324–342.

[12] Lengauer, T. (1990): VLSI Theory. In: *Handbook of Theoretical Computer Science*, Vol. A, pp. 835–868.

[13] Lipton, R. J. and Sedgewick, R. (1981): Lower bounds for VLSI. In *Proc. of 13th Ann. ACM Symp. on the Theory of Computing*, 300–307.

[14] Nisan, N. (1991): CREW PRAMs and decision trees, *SIAM J. Comput.* **20**:6, 999–1007.

[15] Papadimitriou Ch. H. and Sipser M. (1982): Communication complexity. In *Proc. of 14th Ann. ACM Symp. on the Theory of Computing*, pp. 196–200. Journal version in: *J. Comput. Syst. Sci.*, **28**:2 (1984), 260–269.

[16] Razborov, A. (1992): On the distributional complexity of disjointness, *Theoretical Comput. Sci.* **106**:2, 385–390.

[17] Sauerhoff, M. (1999): Lower bounds for randomized read-k-times branching programs. In *Proc. of 15th Symp. on Theoretical Aspects in Comput. Sci.*, Springer Lecture Notes in Computer Science **1373**, 105–115.

[18] Sauerhoff, M. (2003): Randomness versus nondeterminism for read-once and read-k branching programs. In *Proc. of 20th Symp. on Theoretical Aspects in Comput. Sci.*, Springer Lecture Notes in Computer Science **2607**, 307–318.

[19] Tardos, G. (1989) Query complexity, or why is it difficult to separate $\mathrm{NP}^A \cap \text{co-NP}^A$ from P^A by a random oracle A? *Combinatorica* **9**, 385-392.

[20] Wegener, I. (2000): *Branching programs and Binary Decision Diagrams: Theory and Applications.* SIAM Series in Discrete Mathematics and Applications.

[21] Yao, A. C. (1979): Some complexity questions related to distributed computing. In *Proc. of 11th Ann. ACM Symp. on the Theory of Computing*, 209–213.

[22] Yao, A. C. (1983): Lower bounds by probabilistic arguments. In *Proc. of 24th Ann. IEEE Symp. on Foundations of Comput. Sci.*, 420–428.