Séparation des modèles de communication simultanée pour les réseaux: protocoles déterministes, avec bits aléatoires publics et privés.[†]

Florent Becker[1], Pedro Montealegre[1], Ivan Rapaport[2,3], Ioan Todinca[1].

[1] *Univ. Orléans, INSA Centre Val de Loire, LIFO EA 4022, Orléans, France*
[2] *Departamento de Ingeniería Matemática, Univ. de Chile, Chile*
[3] *Centro de Modelamiento Matemático (UMI 2807 CNRS), Univ. de Chile, Chile*

Nous étudions un modèle de communication à plusieurs joueurs, où les joueurs correspondent aux noeuds d'un réseau. Chaque joueur connaît la taille du réseau, son identifiant ainsi que ceux de ses voisins. Les joueurs envoient simultanément un unique message à un arbitre, qui doit décider une propriété du graphe. L'objectif de l'article est de séparer, du point de vue de la complexité de communication (la taille des messages), trois situations différentes : les protocoles déterministes, les protocoles probabilistes avec des bits aléatoires publics et les protocoles probabilistes avec uniquement des bits aléatoires privés. Pour ce faire nous travaillons autour de la fonction booléenne Jumeaux, dont le résultat est *vrai* si le graphe possède deux sommets ayant le même voisinage, et *faux* sinon.

Keywords: Algorithmes distribués; Complexité de communication; Protocoles randomisés

1 Introduction

In the *number-in-hand* multiparty communication model there are k players. Each of these k players receives an n-bit input string x_i and they all need to collaborate in order to compute some function $f(x_1,\dots,x_k)$. There are different communication modes for the *number-in-hand* model. In this paper we focus on the *simultaneous messages* communication mode, in which all players simultaneously send a unique message to a referee. The referee collects the messages and computes the function f. The computational power of both the players and the referee is unlimited. When designing a protocol for function f, the goal is to minimize the size of the longest message generated by the protocol. This minimum, usually depending on n, is called the *message size complexity* of f. Typical questions in communication complexity consist in designing protocols with small messages, and proving lower bounds on the size of such messages.

Several authors considered the case where the data distributed among the players is a graph [AGM12, BMN$^+$11, PVZ12, WZ13]. Informally, each player knows a set of edges of the graph and together they must decide a graph property, e.g., connectivity. Again we can observe two different settings. In one of them, the edges are distributed among the players in an adversarial way [AGM12, WZ13]. In this work, following [AGM12, BMN$^+$11], we consider the setting where each player corresponds to a node of the graph, and thus each player knows the identifier of this node together with the identifiers of its neighbors and the size of the graph, represented as an n-bits vector (in the vector x_i of player i, the bit number j is set to 1 if and only if the nodes i and j are adjacent). For the sake of simplicity we assume that the graph has n nodes numbered from 1 to n, hence there are $k = n$ players, and we call this model *number-in-hand for networks*.

[†] Une version étendue de ces travaux a été acceptée à SIROCCO'14. This work has been partially supported by CONICYT via Basal in Applied Mathematics (I.R.), Núcleo Milenio Información y Coordinación en Redes ICM/FI P10-024F (I.R.) and Fondecyt 1130061 (I.R.)

For many natural functions the messages are much shorter when randomization is allowed [KN97]. In the randomized setting, there are significant differences between the communication complexities of protocols using *public coins* (shared by all players and the referee) and the more restrictive setting where each player has his own, *private coin*. We emphasize that in the *number-in-hand communication model for networks*, each edge is "known" by two players, thus we have some shared information.

Related work.

The number-in-hand model with simultaneous messages and $k = 2$ players.
The case of two players is not new and it has been intensively studied. Clear separations have been proved between deterministic, private coins and public coins protocols in this case. For instance, the message size complexity of the EQ function, which simply tests whether the two n-bit inputs are equal, is $\Theta(n)$ for deterministic protocols [KN97], $O(1)$ for randomized protocols with public coins with constant one-sided error [BK97], and $\Theta(\sqrt{n})$ for randomized protocols with private coins and constant one-sided error [BK97] (see Section 2 for details). More generally, Babai and Kimmel [BK97] proved that for any function f its randomized message size complexity, for private coins protocols, is at least the square root of its deterministic message size complexity. Chakrabarti *et al.* [CSWY01] proved that, for some family of functions, the gap between deterministic and randomized message size complexity with private coins is smaller that the square root.

The number-in-hand communication model for networks.
For deterministic protocols, Becker *et al.* [BMN+11] show that graphs of bounded degeneracy can be completely reconstructed by the referee using messages of size $O(\log n)$, and several natural problems like deciding whether the graph has a triangle, or if its diameter is at most 3, have message size complexity of $\Theta(n)$. For randomized protocols with public coins, Ahn, Guha and McGregor [AGM12] introduced a beautiful and powerful technique for *graph sketching*. The technique works both for streaming models and for the *number-in-hand for networks*, and allows to solve CONNECTIVITY using messages of size $O(\log^3 n)$. The protocols have two-sided, $O(1/n^c)$ error, for any constant $c > 0$.

Our results. We separate the deterministic, the randomized with private coins and the randomized with public coins settings of the *number-in-hand for networks* communication model. The separations are made using problem TWINS and some variants. The boolean function TWINS(G) returns 1 if and only if graph G has two twins (that is, two nodes having the same neighborhood). We also consider function TWIN$_x(G)$, where x is the identifier of a node, and the result is 1 if and only if there is some other node having the same neighborhood as x.

We prove that the deterministic message size complexity of TWINS and TWIN$_x$ is $\Theta(n)$. Also, both functions can be computed by randomized protocols with public coins and message size $O(\log n)$. These protocols, based on the classical fingerprint technique, have one-sided error $O(1/n^c)$ for any constant $c > 0$. Observe that the situation for private coins is very different from the case of the *number-in-hand* model with two players, where the gap between private coins and determinism is at most the square root. In order to separate the private and public coins settings we use a boolean function called TRANSLATED-TWINS. We prove that the message size complexity of this function in the private coins setting is $\Omega(\sqrt{n})$, while it is $O(\log n)$ in the public coins setting. The main results of this research are summarized in Table 1.

	TWINS	TWIN$_x$	TRANSLATED-TWINS
Deterministic	$\Theta(n)$	$\Theta(n)$	$\Theta(n)$
Randomized private-coins	$O(\sqrt{n}\log n)$	$O(\log n)$	$\Omega(\sqrt{n})$, $O(\sqrt{n}\log n)$
Randomized public-coins	$O(\log n)$	$O(\log n)$	$O(\log n)$

TABLE 1: Main results of this research.

2 Definitions

Number-in-hand. The *number-in-hand* communication model is defined as follows. Let f be a function having as input k boolean vectors of length n. There are k players $\{p_1, \dots, p_k\}$ who wish to compute the value of f on input $(x_1, \dots, x_k) \in (\{0,1\}^n)^k$. Player p_i only sees the input x_i, and also knows his own number i. We only consider here the *simultaneous messages* communication mode, in which all the k players simultaneously send a message to a *referee*. After that, the referee (another player who sees none of the inputs) announces the value $f(x_1,\dots,x_k)$ using only the information contained in the k messages.

A *deterministic protocol* $\mathcal{P}$ for function f describes the algorithms of the players (for constructing the messages) and of the referee (for retrieving the final result) that correctly computes f on all inputs. An ε-*error randomized protocol* $\mathcal{P}$ for f is a protocol in which every player and the referee are allowed to use a sequence of random bits, and for all $(x_1,\dots,x_k) \in (\{0,1\}^n)^k$ the referee outputs $f(x_1,\dots,x_k)$ with probability at least $1-\varepsilon$. For boolean functions f we define a *one-sided* ε-*error randomized protocol* in the same way, with exception that for all $(x_1,\dots,x_k) \in (\{0,1\}^n)^k$ such that $f(x_1,\dots,x_k) = 1$, the referee always outputs 1. We distinguish between two sub-cases of randomized protocols : (i) the *private-coin* setting, in which each player, including the referee, flips private coins and (ii) the *public-coin* setting, where the coins are shared between players, but the referee can still have his own private coins. The *cost* of a protocol $\mathcal{P}$, denoted OREQ, is the length of the longest message sent to the referee. The *deterministic message size complexity*, denoted $C^{\text{det}}(f)$, is the minimum cost of any deterministic protocol computing f. Analogously, we denote $C_\varepsilon^{\text{priv}}(f)$, $C_\varepsilon^{\text{pub}}(f)$, as the message size complexity for ε-error public and private protocols, respectively.

Number-in-hand for networks. *Number-in-hand for networks* is a particular case of *number-in-hand* where each party is a node of an n-vertex graph with vertices numbered from 1 to n. Therefore, in this model, $k = n$, player p_i corresponds to the node i and the inputs $x_1,\dots,x_n$ correspond to the rows of the adjacency matrix of some simple undirected graph G of size n. Hence, the input of player (node) i is the characteristic function of the neighborhood $N_G(i)$ (i.e. $j \in N_G(i)$ if and only if $ij \in E(G)$).

All our graphs are undirected, so for any pair i, j of nodes, the bit number i of player j equals the bit number j of player i. In full words, each edge of the graph is known by the two players corresponding to its end-nodes. All our protocols use $\Omega(\log n)$ bits. We assume, w.l.o.g., that each node sends its own number in the message transmitted to the referee.

The problems. We now come back to the *number-in-hand for networks* model. In this framework we shall study three boolean functions on graphs.

- TWINS(G) outputs 1 if and only if G has two vertices u and v with the same neighborhood, i.e., such that $N(u) = N(v)$.
- TWINS$_x(G)$ is a “pointed” version of previous function. Its output is 1 if and only if there is a vertex y such that $N(y) = N(x)$.
- TRANSLATED-TWINS is defined on input graphs G of size $2n$, labeled from 1 to $2n$. Its output is 1 if and only if G has a vertex i such that, for any vertex j, $j \in N(i) \iff j+n \in N(i+n)$. In other words, the output is 1 if and only if there exists i such that $N(i)+n = N(i+n)$.

3 Lower bounds in the deterministic model

The upper bounds of $O(n)$ are trivial so we only need to prove the lower bounds. To show the upper bounds in the deterministic model (the first row in Table 1) we combine two ingredients. First of all, consider the function RECONSTRUCTION(G), whose output is G itself, i.e., the adjacency matrix of G. Note that if a deterministic protocol computes RECONSTRUCTION on the family of n-vertex graphs $\mathcal{G}_n$, then such protocol must generate messages of size at least $\log(|\mathcal{G}|)/n$ (see also [BMN$^+$11]).

Then, we show that if we have a deterministic protocol $\mathcal{P}$ for any of the three problems (TWINS, TWIN$_x$ or TRANSLATED-TWINS), then we can use this protocol to build another protocol $\mathcal{P}'$ for RECONSTRUCTION, such that the cost of $\mathcal{P}'$ is roughly the cost of $\mathcal{P}$ times a constant. The construction of $\mathcal{P}'$ from $\mathcal{P}$ is almost identical for TWINS and TWIN$_x$, but is quite different in the case of TRANSLATED-TWINS. More details in [BMRT14].

4 Upper bounds in the randomized model

The lower bounds obtained for randomized models are based in the well known technique called *fingerprints*, used in the case of 2 players. Consider the problem EQ mentioned in the introduction, and name x_1 and x_2 the inputs of the players, which are numbers in $[0, 2^n]$. The fingerprint technique consists in taking a prime number p uniformly at random, and then player i sends $f_i = x_i \mod p$ to the referee, who simply has to check if $f_1 = f_2$. Why does this work ? If $x_1 = x_2$ then the fingerprint f_1 equals f_2. By the other hand, if $x_1 \neq x_2$, it is easy to check that if we pick $p \in [n^3, 2n^3]$ the probability that $f_1 = f_2$ is smaller than $1/n$. Notice that if we take a bigger prime number p, say in $[n^4, 2n^4]$ then the (one-sided) error probability of the protocol is reduced to smaller $1/n^2$. Then, back in the n players case and following the same technique, we can solve TWINS, TWIN$_x$ and TRANSLATED-TWINS with error probability smaller than $1/n$.

The fingerprint technique requires that the random prime number p is equal for each player, which means that this technique works only in the public-coins randomized model. However, in the case of TWIN$_x$ it is possible to fix this issue. Indeed, suppose that player i has its input x_i and its own randomized prime number p_i. Since x_i is the neighborhood of i in G, i can recognize if x is in its neighborhood or not. Then, a protocol in this context could be : make a fingerprint f_i with p_i, that is $f_i = x_i \mod p_i$, and send f_i together with p_i and one more bit, $0-1$ depending if x is in the neighborhood of i or not. The referee first recover x from the last bit of each message, for each i compute $f_x^i = x \mod p_i$, and test if $f_i = f_x^i$. Choosing p big enough, we obtain the same costs and errors.

5 Lower bounds in the private-coins randomized model

The results shown before clearly separate the deterministic model from the private and public randomized models. To separate the two randomized models, we show that TRANSLATED-TWINS requires $\Omega(\sqrt{n})$ bits of communication in the private coins randomized model. The proof is based in the result of Chakrabarti *et al.* [CSWY01]. Let OREQ be the problem where each player receives a squared matrix of size n, and the output is 1 if there is a index $i \in [1, n]$ such that the i-th row of both matrices are equal. In [CSWY01], it is shown that any protocol solving OREQ requires $\Omega(n\sqrt{n})$ bits of communication in the private coins randomized model for two players. We reduced TRANSLATED-TWINS to OREQ, showing that if there exists a protocol $\mathcal{P}$ for TRANSLATED-TWINS then there is a protocol for OREQ with cost $O(nC(\mathcal{P}))$. That shows than the cost of TRANSLATED-TWINS is $\Omega(\sqrt{n})$ for any public coins randomized protocol.

6 Open problems

The first natural challenge is to determine the message size complexity of function TWINS for randomized protocols with private coins. Using the techniques of Babai and Kimmel [BK97] for EQ, one can prove that TWINS can be solved by a one-sided, bounded error protocol with private coins and messages of size $O(\sqrt{n}\log n)$. We believe that the message size complexity of TWINS for private coins protocols is $\Omega(\sqrt{n})$.

More surprisingly, to the best of our knowledge, the message size complexity of CONNECTIVITY is wide open. Recall that, in the randomized, public coins setting, there exists a protocol using $O(\log^3 n)$ bits, due to Ahn, Guha and McGregor [AGM12]. Can this upper bound be improved to $O(\log n)$? For randomized protocols with private coins and/or for deterministic protocols, can one prove a lower bound of $\Omega(n^c)$ for some constant $c < 1$?

Références

[AGM12] Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. Analyzing graph structure via linear measurements. In *Proc. of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 459–467, 2012.

[BK97] László Babai and Peter G. Kimmel. Randomized simultaneous messages : Solution of a problem of Yao in communication complexity. In *Proc. of the 12th Annual IEEE Conference on Computational Complexity*, pages 239–246, 1997.

[BMN$^+$11] Florent Becker, Martín Matamala, Nicolas Nisse, Ivan Rapaport, Karol Suchan, and Ioan Todinca. Adding a referee to an interconnection network : What can(not) be computed in one round. In *Proc. of the 25th IEEE International Parallel and Distributed Processing Symposium*, IPDPS '11, pages 508–514, 2011.

[BMRT14] Florent Becker, Pedro Montealegre, Ivan Rapaport, and Ioan Todinca. The simultaneous number-in-hand communication model for networks : Private coins, public coins and determinism. In *Structural Information and Communication Complexity*, volume 8576 of *Lecture Notes in Computer Science*, pages 83–95. Springer International Publishing, 2014.

[CSWY01] A. Chakrabarti, Yaoyun Shi, A. Wirth, and A. Yao. Informational complexity and the direct sum problem for simultaneous message complexity. In *Proc. of the 42nd IEEE Symposium on Foundations of Computer Science*, FOCS '01, pages 270–278, 2001.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, New York, NY, USA, 1997.

[PVZ12] Jeff M. Phillips, Elad Verbin, and Qin Zhang. Lower bounds for number-in-hand multiparty communication complexity, made easy. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 486–501, 2012.

[WZ13] David P. Woodruff and Qin Zhang. When distributed computation is communication expensive. In *Proc. of the 27th International Symposium on Distributed Computing*, volume 8205 of *Lecture Notes in Computer Science, DISC '13*, pages 16–30, 2013.