The simultaneous number-in-hand communication model for networks: private coins, public coins and determinism*

Florent Becker[1], Pedro Montealegre[1], Ivan Rapaport[2,3], and 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

Abstract. We study the multiparty communication model where players are the nodes of a network and each of these players knows his/her own identifier together with the identifiers of his/her neighbors. The players simultaneously send a unique message to a referee who must decide a graph property. The goal of this article is to separate, from the point of view of message size complexity, three different settings: deterministic protocols, randomized protocols with private coins and randomized protocols with public coins. For this purpose we introduce the boolean function TWINS. This boolean function returns 1 if and only if there are two nodes with the same neighborhood.

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 $f(x_1, \ldots, x_k)$. Despite its simplicity, the case $k > 2$ started to be studied very recently [1, 2, 4, 6–8, 13, 14].

There are different communication modes for the *number-in-hand* model. In this paper we focus on the *simultaneous message* 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 [1, 4, 13, 14]. Informally, each player knows a set of edges of the graph and together they must decide a graph property, e.g., connectivity. Again

* 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.)

we can observe two different settings. In one of them, the edges are distributed among the players in an adversarial way [1, 14]. In this work, following [1, 4], 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, 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*.

For many natural functions the messages are much shorter when randomization is allowed [12]. 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. Not surprisingly, as pointed out in [14], this model is stronger than the one where edges are distributed in an adversarial way among players.

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 [12], $\mathcal{O}(1)$ for randomized protocols with public coins with constant one-sided error [3], and $\Theta(\sqrt{n})$ for randomized protocols with private coins and constant one-sided error [3] (see Section 2 for details). More generally, Babai and Kimmel [3] 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.* [5] 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.* [4] show that graphs of bounded degeneracy can be completely reconstructed by the referee using messages of size $\mathcal{O}(\log n)$, 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 [1, 2] 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 $\mathcal{O}(\log^2 n)$. The protocols have two-sided, $\mathcal{O}(1/n^c)$ error, for any constant $c > 0$.

Our results. In this paper 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 $\mathcal{O}(\log n)$. These protocols, based on the classical fingerprint technique, have one-sided error $\mathcal{O}(1/n^c)$ 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 (see Section 2 for details). We prove that the message size complexity of this function in the private coins setting is $\Omega(\sqrt{n})$, while it is $\mathcal{O}(\log n)$ in the public coins setting. The main results of this paper are summarized in Table 1.

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

There are several natural problems that cannot be solved with randomized protocols using $o(n)$ bits. In the last part of this paper (Theorem 5) we sketch how the arguments of [4], for proving negative results on deterministic protocols, can be extended to the randomized setting. More precisely, we prove that the randomized public coin message size complexity of the boolean functions TRIANGLE(G) (that outputs 1 if and only if G has a triangle) and DIAM3(G) (that outputs 1 if and only if G has diameter at most 3) is $\Omega(n)$.

2 Preliminaries

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 $\{p_1, \ldots, p_k\}$ who wish to compute the value of f on input $(x_1, \ldots, x_k) \in (\{0,1\}^n)^k$. 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 $f(x_1, \ldots, x_k)$ 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 $(x_1, \ldots, x_k) \in (\{0,1\}^n)^k$ the referee outputs $f(x_1, \ldots, x_k)$ with probability at least $1 - \epsilon$. For boolean functions f we define a *one-sided ϵ-error randomized protocol* in the same way, with exception that for all $(x_1, \ldots, x_k) \in (\{0,1\}^n)^k$ such that $f(x_1, \ldots, x_k) = 1$, 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 $C(\mathcal{P})$, is the length of the longest message sent to the referee. The *deterministic message size complexity*, denoted $C^{\det}(f)$, is the minimum cost of any deterministic protocol computing f. Analogously, we denote $C^{\mathrm{priv}}_{\epsilon}(f)$, $C^{\mathrm{pub}}_{\epsilon}(f)$, 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 $x_1, \ldots, x_n$ 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.

Known results. Let us recall some classical results of the *number-in-hand* model with two players. Babai and Kimmel [3] have shown that the order of magnitude of the private-coins randomized message size complexity of any function f is at least the square root of the deterministic message size complexity of f. They also characterize completely the function: $\mathrm{EQ} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, where $\mathrm{EQ}(x, y) = 1$ iff $x = y$.

Proposition 1 ([3]). *Consider the number-in-hand model with two players and a constant $\epsilon > 0$. The function* EQ *on two n-bit boolean vectors has the following message size complexities: $C^{\det}(\mathrm{EQ}) = n$, $C^{\mathrm{priv}}_{\epsilon}(\mathrm{EQ}) = \Theta(\sqrt{n})$ and $C^{\mathrm{pub}}_{\epsilon}(\mathrm{EQ}) = \mathcal{O}(1)$. For any boolean function f, $C^{\mathrm{priv}}_{\epsilon}(f) = \Omega(\sqrt{C^{\det}(f)})$.*

We also use the following result of Chakrabarti *et al.* [5] for private coins protocol; the deterministic part is a matter of exercise.

Proposition 2 ([5]). *Consider the boolean function* OREQ *that takes as input two boolean $n \times n$ matrices, and the output is 1 if and only if there is some*

$1 \leq i \leq n$ *such that the* i*-th lines of the two matrices are equal. Then, for any* $\epsilon < 1/2$, $C^{\text{priv}}_{\epsilon}(\text{OREQ}) = \Omega(n\sqrt{n})$. *Also,* $C^{\text{det}}(\text{OREQ}) = \Theta(n^2)$.

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 $N(i) + n = N(i + n)$.

For reductions we also use 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 [4]).

3 Deterministic protocols

Theorem 1. *The deterministic message size complexity of functions* TWINS, TWIN$_x$ *and* TRANSLATED-TWINS *is* $\Theta(n)$.

The upper bounds of $\mathcal{O}(n)$ are trivial so we only need to prove the lower bounds. For the first two problems, we use the following reduction.

Lemma 1. *Assume that there is a deterministic protocol solving* TWINS *(resp.* TWINS$_{2n+1}$*) on* $2n + 1$*-node graphs using messages of size* $g(n)$. *Then one can solve* RECONSTRUCTION *on* n*-node graphs using messages of size* $2g(n)$.

Proof. Let G be an arbitrary n-nodes graph, i be an integer between 1 and n and S be a subset of $\{1, \ldots, n\}$ not containing i. Denote by $H(i, S)$ the graph on $2n + 1$ nodes obtained as follows (see Figure 1):

1. $H[\{1, \ldots, n\}] = G$.
2. For each $n + 1 \leq j \leq 2n$, its unique neighbor with identifier at most n is $j - n$.
3. Node $2n + 1$ is adjacent exactly to the nodes of S and to $i + n$.

Claim. We claim that TWINS$(H(i, S))$ (resp. TWINS$_{2n+1}(H(i, S))$ is true if and only if $N_G(i) = S$.

Clearly, if $N_G(i) = S$ then node i is a twin of $2n+1$ in graph H. Conversely, we prove that if $H(i, S)$ has two twins u and v then one of them is $2n + 1$. This comes from the fact that the edges between $\{1, \ldots, n\}$ and $\{n+1, \ldots, 2n\}$ in $H(i, S)$ form a matching, so no two nodes of $\{1, \ldots, 2n\}$ may be twins. Now

assume that $2n+1$ has a twin u. Since $N_{H(i,S)}(2n+1) \cap \{n+1, \ldots, 2n\} = \{i+n\}$, the only possibility is that $u = i$. Eventually, i and $2n+1$ are twins if and only if $N_G(i) = S$, which proves our claim.

Now assume that we have a distributed protocol for TWINS (or TWINS_{2n+1}) on graphs with $2n+1$ nodes (actually it suffices to consider graphs from the family H described above). We construct an algorithm for RECONSTRUCTION on an arbitrary n-nodes graph G.

The players construct their messages as follows. Each node i sends the message m_i that it would send in the TWINS protocol if it had neighborhood $N_G(i) \cup \{i+n\}$ and the message m_i^+ that it would send in the same protocol with neighborhood $N_G(i) \cup \{i+n, 2n+1\}$. That makes messages of size $2g(n)$.

The referee needs to retrieve the neighborhood $N_G(i)$ for each i, from the set of messages. For each i and each subset S of $\{1, \ldots, n\}$ not containing i, we simulate the behavior of the referee for TWINS on graph $H(i,S)$. For this purpose, for each $j \leq n$ we use message m_j if $j \notin S$ and message m_j^+ if $j \in S$. The messages for nodes $k > n$ can be constructed directly by the referee. Note that $\text{TWINS}(H(i,S))$ is true iff $N_G(i) = S$, thus we can reconstruct $N_G(i)$. Eventually, this allows to solve RECONSTRUCTION on graph G. The same arguments work of we replace the TWINS protocol by TWINS_{2n+1}. □

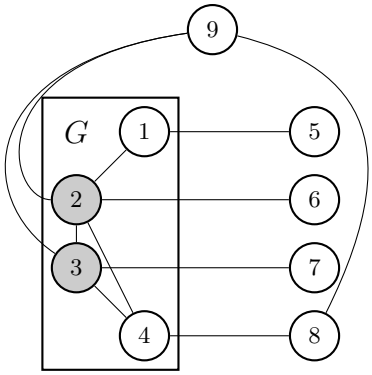

Fig. 1. $H(4,S)$, when $S = \{2,3\}$

Remark 1. Since problem RECONSTRUCTION on n-node graphs requires messages of size $\Omega(n)$, we conclude that any deterministic protocol for either TWINS or TWINS_{2n+1} also requires messages of size $\Omega(n)$.

For problem TRANSLATED-TWINS, we provide a reduction from OREQ (see Proposition 2 in Section 2). It will be used both for deterministic and randomized protocols with private coins.

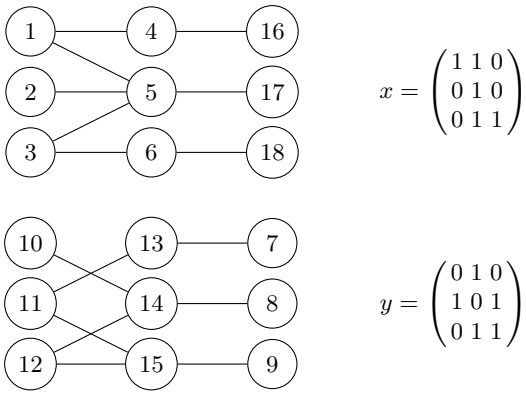

Fig. 2. Examples of graphs G_x^1 (top) and G_y^2 (bottom), for a given input (x, y). This is a *yes* instance since $x_3 = y_3$.

Lemma 2. *Assume that there is a protocol solving* Translated-Twins *for* $6n$*-node graphs using messages of size* $g(n)$*, in any of our three settings. Then there is a protocol for function* OREQ*, in the same setting, using messages of size* $3ng(n)$*.*

Proof. Let x and y be two $n \times n$ boolean matrices. We construct a graph $G_{x,y}$ with $6n$ nodes such that $\text{Translated-Twins}(G_{x,y}) = \text{OREQ}(x, y)$.

The graph G is formed by two connected components G_x^1 and G_y^2 of $3n$ nodes each, encoding the two matrices as follows (see Figure 2 for an example).

G_x^1 has $3n$ nodes numbered from 1 to $2n$ and from $5n + 1$ to $6n$. For any $i, j \in \{1, \ldots, n\}$ we put an edge between node i and node $j + n$ if and only if $x_{i,j} = 1$. Then for any $i \in \{1, \ldots, n\}$ we put an edge between node $i+n$ and node $i + 4n$. In other words, the node subsets $\{1, \ldots, n\}$ and $\{n+1, \ldots, 2n\}$ induce a bipartite graph representing matrix x, and the node subsets $\{n+1, \ldots, 2n\}$ and $\{5n + 1, \ldots, 6n\}$ induce a perfect matching.

The construction of G_y^2, with nodes numbered from $2n+1$ to $5n$ is similar. For any $i, j \in \{1, \ldots, n\}$ we put an edge between node $i + 3n$ and node $j + 4n$ if and only if $y_{i,j} = 1$. Also, for any $i \in \{1, \ldots, n\}$, we put an edge between node $4n + i$ and node $2n + i$. Thus the node subsets $\{3n + 1, \ldots, 4n\}$ and $\{4n + 1, \ldots, 5n\}$ form a bipartite graph corresponding to matrix y. The subsets $\{4n + 1, \ldots, 5n\}$ and $\{2n + 1, \ldots, 3n\}$ induce a matching.

We claim that $\text{Translated-Twins}(G_{x,y}) = \text{OREQ}(x, y)$. Assume that $\text{OREQ}(x, y) = 1$. There is an index i such that line number i in x equals line number i in y. Then, by construction, the neighborhood of node $i + 3n$ in $G_{x,y}$ is the neighborhood of node i, translated by an additive term $3n$.

Conversely, assume that there is some node $u \in \{1, \ldots, 3n\}$ such that the neighborhood of u is the translated neighborhood of $u + 3n$. By construction, the only possibility is that $u \leq n$ (because of the numberings of the matchings

the other nodes cannot have translated twins), thus line number u is the same in the two matrices.

To achieve the proof of our lemma, assume that we have a protocol for TRANSLATED-TWINS for graphs with $3n$ nodes, with $g(n)$ bits per message. We design a protocol for OREQ. Recall that for OREQ, each player has a matrix, say x for the first one and y for the second one. The first player constructs graph $G_{x,0} = (G_x^1, G_0^2)$, the second constructs $G_{0,y} = (G_0^1, G_y^2)$ (here 0 denotes the $n \times n$ boolean matrix whose elements are all 0). The first player sends the $3n$ messages corresponding to the nodes of G_x^1 in the TRANSLATED-TWINS protocol for graph $G_{x,0}$. The second player sends the $3n$ messages corresponding to the nodes of G_y^2 in protocol TRANSLATED-TWINS for $G_{0,y}$. The referee collects these $6n$ messages; observe that they are exactly those sent by protocol TRANSLATED-TWINS for the graph $G_{x,y}$. He applies the same algorithm as the referee of TRANSLATED-TWINS on these messages. By the claim above, its output is TRANSLATED-TWINS$(G_{x,y})$, thus OREQ(x, y). Note that the messages used here are of size $\mathcal{O}(3ng(n))$ and that our arguments hold for any type of protocol. □

This achieves the proof of Theorem 1.

4 Randomized protocols

Theorem 2. *For any constant $c > 0$,* TWINS, TWINS$_x$ *and* TRANSLATED-TWINS *can be solved by randomized protocols with public coins using messages of size $O(\log n)$ and $1/n^c$ one-sided error. Problem* TWINS$_x$ *can also be solved by a randomized protocol with private coins using messages of size $O(\log n)$ and $1/n^c$ one-sided error.*

Proof. Let $n^{c+3} < p \leq 2n^{c+3}$ be a prime number. A random $t \in \mathbb{Z}_p$ is chosen uniformly at random using $\mathcal{O}(\log(n))$ public random bits. Given an n-bits vector $a = (a_1, \ldots, a_n)$, consider the polynomial $P_a = a_1 + a_2 X + a_3 X^2 + \ldots a_n X^{n-1}$ in $Z_p[X]$ and let $FP(a, t) = P_a(t)$. $FP(a, t)$ is sometimes called the "fingerprint" of vector a. Clearly two equal vectors have equal fingerprints, and, more important, for any two different vectors a and b, the probability that $FP(a, t) = FP(b, t)$ is at most $1/n^{c+2}$ (because the polynomial $P_a - P_b$ has at most n roots and t was chosen uniformly at random, thus the probability that t is a root of $P_a - P_b$ is at most $1/n^{c+2}$, see e.g., [11]).

Let x_i be the input vector of player (node) number i, i.e., the characteristic function of its neighborhood $N(i)$. A protocol for TWINS consists in each node sending the message $m_i = FP(x_i, t)$. The referee outputs 1 if and only if $m_i = m_j$ for some pair $i \neq j$. A protocol for TWINS$_x$ send the same messages, but this time the referee checks whether $m_x = m_i$ for some $i \neq x$. The protocol for TRANSLATED-TWINS on n-node graphs is slightly different. If a node $i \leq n/2$ has a neighbor $j > n/2$, it sends a special "no" message specifying that it cannot be a candidate for having a translated twin. Otherwise, let y_i^1 be the $n/2$-bits vector formed by the $n/2$ first bits of x_i. Thus y_i^1 is the characteristic vector of $N(i) \cap \{1, \ldots, n/2\}$. Player i sends the message $m_i = FP(y_i^1, t)$. Symmetrically,

for nodes labelled $i > n/2$, if i has some neighbor $j \leq n/2$ it sends the "no" message. Otherwise, let y_i^2 be the $n/2$-bits vector formed by the last $n/2$ bits of x_i. Hence y_i^2 corresponds to $N(i) \cap \{n/2, \ldots, n\}$, "translated" by $-n/2$. Player i sends the message $m_i = FP(y_i^2, t)$. Then the referee returns 1 if $m_i = m_{i+n/2}$ for some $i \leq n/2$.

Clearly, for protocol TWINS (resp. TWINS$_x$, TRANSLATED-TWINS), if the input graph is a yes-instance then the protocol outputs 1. The probability that TWINS answers 1 on a no-instance is the probability that $FP(m_i, t) = FP(m_j, t)$ for two nodes i and j with different neighborhoods. For each fixed pair of nodes this probability is at most $1/n^{c+2}$, so altogether the probability of a wrong answer is at most $1/n^c$. With similar arguments for TWINS$_x$ and TRANSLATED-TWINS the probability of a wrong answer is at most $1/n^{c+1}$, since the referee makes n tests and each may be a false positive with probability at most $1/n^{c+2}$.

For TWINS$_x$ with private coins, each node i sends a bit stating if it sees x, a number t_i chosen uniformly at random in the interval $n^{c+2} < p \leq 2n^{c+2}$ and also $FP(x_i, t_i)$. The referee retrieves the neighborhood of node x (which was sent bit by bit by all the others) and then, for each $i \neq x$, it constructs $FP(x_x, t_i)$ and compares it to $FP(x_i, t_i)$. If the values are equal for some i, the referee outputs 1, otherwise it outputs 0. Again any yes-instance will answer 1, and the probability that a no-instance (wrongly) answers 1 is at most $1/n^c$. □

The fact that TRANSLATED-TWINS requires $\Omega(\sqrt{n})$ bits per node for any private coins, ϵ-error randomized protocol follows directly by Lemma 2 and Proposition 2.

Theorem 3. *For any $\epsilon < 1/2$, C_ϵ^{priv}(TRANSLATED-TWINS) $= \Omega(\sqrt{n})$.*

Theorems 2 and 3 show that problem TRANSLATED-TWINS separates the private coins and the public coins protocols.

In order to complete the table of the Introduction, we also observe that problems TWINS$_x$ and TRANSLATED-TWINS can be solved by randomized private coins protocols using $\mathcal{O}(\sqrt{n}\log n)$ bits.

Theorem 4. *For any $c > 0$, there is a randomized private coins protocol for* TWINS *and* TRANSLATED-TWINS *using messages of size $\mathcal{O}(\sqrt{n}\log n)$ and having $1/n^c$ one-sided error.*

Proof. Babai and Kimmel in [3] propose a private coins protocol with 1/3 one sided error and $\mathcal{O}(\sqrt{n})$ communication cost for EQ_n, in the *number-in-hand* model with two players (see Proposition 1). Let us call this protocol $\mathcal{P}_0$. As the authors point out, this protocol is symmetrical, in the sense that both players compute the same function on their own input. We define the protocol $\mathcal{P}$ as one obtained by simulating $(c+2)\log_3 n$ calls to protocol $\mathcal{P}_0$. More formally, in $\mathcal{P}$ each player creates $(c+2)\log_3 n$ times the message that it would create in $\mathcal{P}_0$, using at each time independent tosses of private coins. The referee answers 1 if and only if the referee of $\mathcal{P}_0$ would have answered 1 on each of the $(c+2)\log_3 n$ pairs of messages. Therefore $\mathcal{P}$ is a private coin randomized protocol for EQ_n with one sided error smaller than $1/n^{c+2}$, and cost $\mathcal{O}(\sqrt{n}\log n)$.

A one sided private coin randomized protocol $\mathcal{P}'$ for TWINS is one where each node plays the role of Alice in $\mathcal{P}$ taking as an input the characteristic function of its neighborhood, and then the referee simulates the role of the referee in $\mathcal{P}$ for each pair of messages. Similarily, a protocol $\mathcal{P}''$ for TRANSLATED-TWINS works as follows: each node i sends "*no*" in the same cases described in the proof of Theorem 2, and otherwise it simulates the role of Alice on input y_i^1 formed by the first $n/2$ bits of x_i, if $i \leq n/2$ or on input y_i^2 formed by the $n/2$ last bits of x_i if $i > n/2$, where x_i is the characteristic function of $N(i)$. The referee then simulates the referee of $\mathcal{P}$ on the messages of i and $i+n$ every time none of them say "*no*".

Since $\mathcal{P}$ has just one sided error, if TWINS (resp. TRANSLATED-TWINS) is *true*, $\mathcal{P}'$ (resp. $\mathcal{P}''$) will always accept. On the other hand, if TWINS (resp. TRANSLATED-TWINS) is *false*, then the probability that $\mathcal{P}'$ (resp. $\mathcal{P}''$) accepts is the probability that $\mathcal{P}$ accepts for at least one pair of vertices, and then the error of $\mathcal{P}'$ (resp. $\mathcal{P}''$) is at most n^2 times (resp. n times) the error of $\mathcal{P}$. We obtain that $\mathcal{P}'$ and $\mathcal{P}''$ have at most $1/n^c$ one sided error, and communication cost $\mathcal{O}(\sqrt{n}\log n)$. □

Consider the boolean function TRIANGLE(G) that outputs 1 if and only if G has a triangle, and the function DIAM3(G), that outputs 1 if and only if G has diameter at most 3. In [4] is shown that the deterministic message sizes of these problems are lower-bounded by $\Omega(n)$, using a reduction from RECONSTRUCTION. However, as seen in Theorem 1, a reduction from RECONSTRUCTION does not imply lower-bounds on the message sizes of randomized protocols.

In the following theorem, we extend the techniques in [4] to reduce the problems TRIANGLE(G) and DIAM3(G) from INDEX, showing that the message sizes of randomized protocols for these problems are also of size $\Omega(n)$.

Theorem 5. *For any $\epsilon < 1/2$, any public coins randomized protocol computing* TRIANGLE(G) *(resp.* DIAM3(G)*) with ϵ two-sided error uses messages of size $\Omega(n)$.*

Proof. Consider the INDEX function in the model *number-in-hand* with two players: the first player, say Alice, has as input an m-bits boolean vector x and the second player, Bob, has an integer $q, 1 \leq i \leq m$. Then INDEX$(x, q) = x_q$, the qth coordinate of Alice's vector. We will use the fact that for any $\epsilon < 1/2$, any public coins randomized protocol for INDEX requires $\Omega(m)$ bits (see, e.g., [9, 10] for a proof). We may assume w.l.o.g. that $m = n^2$.

In [4], Becker *et al.* show that for the deterministic communication cost for TRIANGLE and DIAM3 is $\Theta(n)$, by showing that if there is a protocol $\mathcal{P}$ of cost c for TRIANGLE or DIAM3, then there is a protocol for RECONSTRUCTION in bipartite graphs of cost $2c$. We slightly modify their proof to obtain a reduction from INDEX.

Let $\epsilon < 1/2$, and $\mathcal{P}$ be a ϵ-error randomized public coins protocol for TRIANGLES on n-nodes graphs, using $c(n)$ bits. We give a protocol for INDEX using $2n \cdot c(2n+1)$ bits.

Let x be an $m = n^2$-bits vector. Let H_x be the bipartite graph with vertex set $\{1, \ldots, 2n\}$, such that for any $1 \leq k, l \leq n$, if $x_{(k-1)n+l} = 1$ then H_x has an edge between nodes k and $l+n$. Consider the family of graphs $H_x(i,j)$ obtained from H_x by adding a node $2n+1$ whose neighbors are nodes i and $j+n$ (for any $1 \leq i, j \leq n$). Observe that $H_x(i,j)$ has a triangle if and only if $x_{(i-1)n+j} = 1$, in which case the triangle is formed by the nodes $\{i, j+n, 2n+1\}$. To simplify the notation we also define the graph $H_x(0,0)$ obtained from H_x by adding an isolated node $2n+1$.

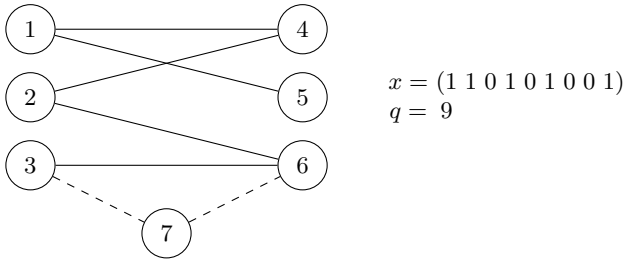

Fig. 3. An illustration of $H_x(3,6)$ when $x = (1,1,0,1,0,1,0,0,1)$ and $q = 9$.

The protocol for INDEX is as follows. Bob sends its input q, which only costs $\mathcal{O}(\log n)$ bits. Alice constructs the family of graphs $H_x(i,j)$, for all pairs $1 \leq i, j \leq n$ and for $(i,j) = (0,0)$. Any node $k \leq 2n$ has exactly two possible of neighborhoods, depending whether it is adjacent to $2n+1$ or not. For each $k \leq 2n$, Alice creates the message $m^+(k)$ that the protocol for TRIANGLE would send for node k in the graph $H_x(k,1)$ (if $k \leq n$) or in the graph $H(1, k-n)$ (if $k > n$). It also creates the message $m^-(k)$ that TRIANGLE would construct for node k in the graph $H_x(0,0)$. In full words, $m^-(k)$ corresponds to the case when the neighborhood of k is the same as in H_x, and $m^+(k)$ to the case when this neighborhood is the neighborhood in H_x, plus node $2n+1$. Then Alice sends, for each k, $1 \leq k \leq 2n$, the pair of messages $(m^-(k), m^+(k))$. Therefore Alice uses $2n \cdot c(2n+1)$ bits. It remains to explain how the referee retrieves the bit x_q. Let i, j such that $q = (i-1)n + j$. Observe that $x_q = 1$ if and only if graph $H_x(i,j)$ has a triangle, therefore the referee must simulate the behavior of the referee for TRIANGLE on $H_x(i,j)$. For this purpose, the referee computes the message that node $2n+1$ would have sent on this graph (it only depends on i and j) and observes that protocol $\mathcal{P}$ on $H_x(i,j)$ would have sent message $m^+(i)$, $m^+(j+n)$ and $m^-(k)$ for any $k \leq 2n$ different from i and j. Therefore the referee can give the same output as $\mathcal{P}$ on $H_x(i,j)$, that is it outputs bit x_q. The protocol for INDEX will have ϵ error and will use $2n \cdot c(2n+1)$ bits. Thus $\mathcal{P}$ requires $\Omega(n)$ bits.

The proof for DIAM3 is based on a similar reduction. Let $D_x(i,j)$ be the graph obtained from H_x by adding three nodes : node $2n+1$ seeing all nodes $k \leq 2n$, node $2n+2$ seeing i and node $2n+3$ seeing $j+n$. Graph $D_x(0,0)$ is

similar with the difference that nodes $2n+2$ and $2n+3$. Observe (see also [4]) that $D_x(i,j)$ has diameter 3 if and only if $x_{(i-1)n+j} = 1$. The rest of the proof follows as before. □

5 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 [3] for EQ, one can prove that Twins can be solved by a one-sided, bounded error protocol with private coins and messages of size $\mathcal{O}(\sqrt{n}\log n)$. 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 $\mathcal{O}(\log^2 n)$ bits, due to Ahn, Guha and McGregor [1]. Can this upper bound be improved to $\mathcal{O}(\log n)$? For randomized protocols with private coins and/or for deterministic protocols, can one prove a lower bound of $\Omega(n^c)$ for some constant $c < 1$?

References

1. K. J. Ahn, S. Guha, and A. McGregor, *Analyzing graph structure via linear measurements*, in Proc. of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '12, 2012, pp. 459–467.
2. ———, *Graph sketches: Sparsification, spanners, and subgraphs*, in Proc. of the 31st Symposium on Principles of Database Systems, PODS '12, 2012, pp. 5–14.
3. L. Babai and P. 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, 1997, pp. 239–246.
4. F. Becker, M. Matamala, N. Nisse, I. Rapaport, K. Suchan, and I. 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, 2011, pp. 508–514.
5. A. Chakrabarti, Y. 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, 2001, pp. 270–278.
6. A. Drucker, F. Kuhn, and R. Oshman, *The communication complexity of distributed task allocation*, in Proc. of the 2012 ACM Symposium on Principles of Distributed Computing, PODC '12, 2012, pp. 67–76.
7. A. Gronemeier, *Asymptotically optimal lower bounds on the NIH-multi-party information complexity of the AND-function and disjointness*, in Proc. of the 26th International Symposium on Theoretical Aspects of Computer Science, STACS '09, 2009, pp. 505–516.
8. T. S. Jaymar, *Hellinger strikes back: A note on the multi-party information complexity of AND*, in Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, vol. 5687 of Lecture Notes in Computer Science, 2009, pp. 562–573.

9. I. Kremer, N. Nisan, and D. Ron, *On randomized one-round communication complexity*, Computational Complexity, 8 (1999), pp. 21–49.
10. ——, *Errata for: "on randomized one-round communication complexity"*, Computational Complexity, 10 (2001), pp. 314–315.
11. E. Kushilevitz, *Communication complexity*, Advances in Computers, 44 (1997), pp. 331–360.
12. E. Kushilevitz and N. Nisan, *Communication Complexity*, Cambridge University Press, New York, NY, USA, 1997.
13. J. M. Phillips, E. Verbin, and Q. 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, 2012, pp. 486–501.
14. D. P. Woodruff and Q. Zhang, *When distributed computation is communication expensive*, in Proc. of the 27th International Symposium on Distributed Computing, vol. 8205 of Lecture Notes in Computer Science, DISC '13, 2013, pp. 16–30.