Exact Classical Simulation of the Quantum-Mechanical GHZ Distribution

Gilles Brassard, Luc Devroye, and Claude Gravel

***Abstract*— John Bell has shown that the correlations entailed by quantum mechanics cannot be reproduced by a classical process involving non-communicating parties. But can they be simulated with the help of bounded communication? This problem has been studied for more than two decades, and it is now well understood in the case of bipartite entanglement. However, the issue was still widely open for multipartite entanglement, even for the simplest case, which is the tripartite Greenberger–Horne–Zeilinger (GHZ) state. We give an exact simulation of arbitrary independent von Neumann measurements on general n-partite GHZ states. Our protocol requires $O(n^2)$ bits of expected communication between the parties, and $O(n \log n)$ expected time is sufficient to carry it out in parallel. Furthermore, we need only an expectation of $O(n)$ independent unbiased random bits, with no need for the generation of continuous real random variables nor prior shared random variables. In the case of equatorial measurements, we improve on the prior art with a protocol that needs only $O(n \log n)$ bits of communication and $O(\log^2 n)$ parallel time. At the cost of a slight increase in the number of bits communicated, these tasks can be accomplished with a constant expected number of rounds.**

***Index Terms*— Entanglement simulation, Greenberger–Horne–Zeilinger (GHZ) state, Knuth-Yao's sampling algorithm, multipartite entanglement, von Neumann's rejection algorithm, random-bit model.**

I. Introduction

THE issue of non-locality in quantum physics was raised in 1935 by Einstein, Podolsky and Rosen when they introduced the notion of entanglement [1]. Thirty years later, Bell proved that the correlations entailed by entanglement cannot be reproduced by classical local hidden variable theories between noncommunicating (e.g. space-like separated) parties [2]. This momentous discovery led to the question of *quantifying* quantum non-locality.

A natural quantitative approach to the non-locality inherent in a given entangled quantum state is to study the amount of resources that would be required in a purely classical theory to reproduce exactly the probabilities corresponding to measuring it. More formally, we consider the problem of *sampling* the joint discrete probability distribution of the outcomes obtained by people sharing this quantum state, on which each party applies locally some measurement on his share. Each party is given a description of his own measurement but is not informed of the measurements assigned to the other parties. This task would be easy (for a theoretician!) if the parties were indeed given their share of the quantum state, but they are not. Instead, they must *simulate* the outcome of these measurements without any quantum resources, using as little *classical communication* as possible. Notice that we insist on *exact* sampling since approximate sampling can obviously be realized by having the parties communicate in order to share approximations of their measurement parameters.

This conundrum was introduced by Maudlin [3] in 1992 in the simplest case of linear polarization measurements at arbitrary angles on the two photons that form a Bell state such as $|\Phi^+\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$. Maudlin claimed that this required "the capacity to send messages of unbounded length", but he showed nevertheless that the task could be achieved with a bounded amount of *expected* communication. Similar concepts were reinvented independently years later by other researchers [4], [5]. This led to a series of results, culminating with the protocol of Toner and Bacon to simulate arbitrary von Neumann measurements on a Bell state with a single bit of communication *in the worst case* [6], thus contradicting Maudlin's claim. Later, Regev and Toner extended this result by giving a simulation of the correlation (but not the marginals) entailed by arbitrary binary von Neumann measurements (meaning that the outcome for each party can take only two values) on arbitrary bipartite states of any dimension using only two bits of communication, also in the worst case [7]. Inspired by Steiner's work [5], Cerf, Gisin and Massar showed that the effect of an arbitrary pair of positive-operator-valued measurements (POVMs) on a Bell state can also be simulated with a bounded amount of expected communication [8]. A more detailed early history of the simulation of quantum entanglement can be found in Ref. [9, Sec. 6].

All this prior work is concerned strictly with the simulation of *bipartite* entanglement. Much less is known when it comes to simulating multipartite entanglement with classical communication, a topic that was still teeming with

Manuscript received May 14, 2015; revised September 22, 2015; accepted October 1, 2015. Date of publication December 1, 2015; date of current version January 18, 2016. G. Brassard was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), in part by the Canada Research Chair Program, and in part by the Institut transdisciplinaire d'information quantique. L. Devroye was supported in part by NSERC and in part by the Fonds de recherche du Québec–Nature et technologies. This paper was presented at the Ninth Conference on Theory of Quantum Computation, Communication, and Cryptography in Singapore, 2014.

G. Brassard is at the Département d'informatique et de recherche opérationnelle (DIRO), Université de Montréal, C.P. 6128, Succursale Centre-Ville, Montreal, Quebec, H3C 3J7 Canada, and with the Canadian Institute for Advanced Research, Toronto, Ontario, Canada. Part of this research was carried out while he was at the Institute for Theoretical Studies of ETH Zürich, Switzerland (e-mail: brassard@iro.umontreal.ca).

L. Devroye is with the School of Computer Science, McGill University, Montreal, Quebec, H3A 0E9, Canada (e-mail: lucdevroye@gmail.com).

C. Gravel is with the Département d'informatique et de recherche opérationnelle, Université de Montréal, Montreal, Quebec, H3C 3J7, Canada (e-mail: claude.gravel@bell.net).

Communicated by M. M. Wilde, Associate Editor for Quantum Information Theory.

Digital Object Identifier 10.1109/TIT.2015.2504525

major open problems. Consider the simplest case, which is the simulation of independent arbitrary von Neumann measurements on the tripartite GHZ state, named after Greenberger, Horne and Zeilinger [10], which we shall denote $|\Psi_3\rangle = \frac{1}{\sqrt{2}}|000\rangle + \frac{1}{\sqrt{2}}|111\rangle$, or more generally on its n-partite generalization $|\Psi_n\rangle = \frac{1}{\sqrt{2}}|0^n\rangle + \frac{1}{\sqrt{2}}|1^n\rangle$.

The easiest situation arises in the special case of *equatorial* measurements (defined in Section II) on the GHZ state because all the marginal probability distributions obtained by tracing out one or more of the parties are uniform. Hence, it suffices in this case to simulate the n-partite correlation. Once this has been achieved, all the marginals can easily be made uniform [11]. Making the best of this observation, Bancal, Branciard and Gisin have given a protocol to simulate equatorial measurements on the tripartite and fourpartite GHZ states at an expected cost of 10 and 20 bits of communication, respectively [12]. Later on, Branciard and Gisin improved this in the tripartite case with a protocol using 3 bits of communication in the worst case [13]. The simulation of equatorial measurements on $|\Psi_n\rangle$ for $n \geq 5$ was handled subsequently by Brassard and Kaplan, with an expected cost of $O(n^2)$ bits of communication [14]. This was the best result obtained until now on this line of work.

Despite substantial effort, the case of *arbitrary* von Neumann measurements, even on the original tripartite GHZ state $|\Psi_3\rangle$, was still wide open. Here, we solve this problem in the general case of the simulation of the n-partite GHZ state $|\Psi_n\rangle$, for any n, under the *random bit model* introduced in 1976 by Knuth and Yao [15], in which the only source of randomness comes from the availability of independently distributed unbiased random bits. Furthermore, we have no needs for prior shared random variables between the parties.[1] Our simulation proceeds with $O(n)$ expected perfect random bits and its expected communication cost is $O(n^2)$ bits, but only $O(n \log n)$ *time* if we count one step for sending bits in parallel according to a realistic scenario in which no party has to send or receive more than one bit in any given step. Furthermore, in the case of equatorial measurements, we improve the earlier best result [14] with an expected communication cost of only $O(n \log n)$ bits and $O(\log^2 n)$ parallel time. At the cost of a slight increase in the number of bits communicated and the number of required random bits, these tasks can be accomplished with a constant expected number of rounds.

More formally, the quantum task that we want to simulate is as follows. Each party j holds one qubit (quantum bit) from state $|\Psi_n\rangle = \frac{1}{\sqrt{2}}|0^n\rangle + \frac{1}{\sqrt{2}}|1^n\rangle$ and is given the description of a von Neumann measurement M_j. By local operations, they collectively perform $\otimes_{j=1}^{n} M_j$ on $|\Psi_n\rangle$, thus obtaining one outcome each, say $b_j \in \{-1, +1\}$, which is their output. The joint probability distribution $p(b)$ of the b_j's is defined by the joint set of measurements. Our purpose is to sample *exactly* this joint probability distribution by a purely classical process that involves no prior shared random variables and as little communication as possible. As mentioned above, previous solutions [12]–[14] required each individual measurement to be equatorial. In order to overcome this limitation, our complete solution builds on three ingredients: (1) Gravel's decomposition of $p(b)$ as a convex combination of two sub-distributions [17], [18]; (2) Knuth and Yao's algorithm [15] to sample exactly discrete probability distributions assuming only a source of unbiased identically independently distributed *bits*, rather than a source of *continuous* uniform random variables on the interval [0, 1]; and (3) our own distributed version of the classic *von Neumann's rejection algorithm* [19].

We define precisely our problem in Section II and we formulate our convex decomposition of the GHZ distribution, which is the key to its simulation. Then, we explain how to sample according to a Bernoulli distribution even when only approximations of the distribution's parameter are available. We also explain how the classic von Neumann rejection algorithm can be used to sample in the sub-distributions defined by our convex decomposition. However, little attention is paid in Section II to the fact that the various parameters that define the joint distribution are not available in a single place. Section III is concerned with the communication complexity issues. This paves the way to Section IV, in which we provide a complete protocol to solve our problem, as well as its detailed analysis. Section V discusses variations on the theme, in which we consider a parallel model of communication, an expected bounded-round solution, improvements on the prior art for the simulation of equatorial measurements, and a remark to the effect that only one party needs access to a source of randomness. We conclude in Section VI with a discussion, open problems, and the announcement of a forthcoming generalization of our results. For completeness, the appendices derive from first principles our convex decomposition of the GHZ distribution, as well as elementary approximation and truncation formulas useful in the analysis of the parallel model.

II. Sampling Exactly the GHZ Distribution in the Random Bit Model

Any von Neumann measurement on a single qubit can be conveniently represented by a point on the surface of a three-dimensional sphere, known as the Bloch sphere, whose spherical coordinates can be specified by an *azimuthal* angle $\theta \in [0, 2\pi)$ and an *elevation* angle $\varphi \in [-\pi/2, \pi/2]$. These parameters define operator

$$M = x\sigma_1 + y\sigma_2 + z\sigma_3 = \begin{pmatrix} \sin\varphi & e^{-\iota\theta}\cos\varphi \\ e^{\iota\theta}\cos\varphi & -\sin\varphi \end{pmatrix},$$

where $x = \cos\theta\cos\varphi$, $y = \sin\theta\cos\varphi$, $z = \sin\varphi$, and σ_1, σ_2 and σ_3 are the Pauli operators. In turn, this operator defines a measurement in the usual way, which we shall also denote M for convenience, whose outcome is one of its eigenvalues $+1$ or -1. A von Neumann measurement is said to be *equatorial* when its elevation angle $\varphi = 0$ vanishes and it is said to be *in the computational basis* when $\varphi = \pm\pi/2$.

Consider a set of n von Neumann single-qubit measurements M_j, represented by their parameters (θ_j, φ_j), where $1 \leq j \leq n$. This set of operators defines a joint measurement

[1] Most of the prior art on the simulation of entanglement by classical communication required the parties to share *continuous* real random variables in an initialization phase [3]–[8], admittedly an unreasonable proposition, but there have been exceptions, such as Ref. [16].

$M = \otimes_{j=1}^{n} M_j$. In turn, this measurement defines a probability distribution p on the set $\{-1,+1\}^n$, which corresponds to the probability of all possible outcomes when the n-partite GHZ state $|\Psi_n\rangle = \frac{1}{\sqrt{2}}|0^n\rangle + \frac{1}{\sqrt{2}}|1^n\rangle$ is measured according to M. We call it the *GHZ distribution*.

Following Refs. [17], [18], we show in Appendix A that the probability $p(b)$ of obtaining $b = (b_1, \ldots, b_n)$ in $\{-1,+1\}^n$ can be decomposed as

$$p(b) = \cos^2\left(\tfrac{\theta}{2}\right) p_0(b) + \sin^2\left(\tfrac{\theta}{2}\right) p_1(b) \tag{1}$$

where

$$\left.\begin{aligned} \theta &= \textstyle\sum_{j=1}^{n} \theta_j \\ p_0(b) &= \tfrac{1}{2}\big(\mathrm{a}_0(b) + \mathrm{a}_1(b)\big)^2 \\ p_1(b) &= \tfrac{1}{2}\big(\mathrm{a}_0(b) - \mathrm{a}_1(b)\big)^2 \end{aligned}\right\} \tag{2}$$

$$\left.\begin{aligned} \mathrm{a}_0(b) &= \textstyle\prod_{j=1}^{n} \cos\left(\tfrac{1}{2}\left(\varphi_j - \tfrac{\pi}{2} b_j\right)\right) \quad \text{and} \\ \mathrm{a}_1(b) &= (-1)^n \textstyle\prod_{j=1}^{n} \sin\left(\tfrac{1}{2}\left(\varphi_j - \tfrac{\pi}{2} b_j\right)\right). \end{aligned}\right\} \tag{3}$$

Hence, we see that distribution p is a convex combination of sub-distributions p_0 and p_1, whose coefficients $\cos^2(\theta/2)$ and $\sin^2(\theta/2)$ depend only on the azimuthal angles of the measurements, whereas the sub-distributions depend only on their elevation angles.

Sampling p is therefore a matter of sampling a Bernoulli distribution with defining parameter $\cos^2(\theta/2)$ before sampling either p_0 or p_1, whichever is the case. As we shall see, full knowledge of the parameters is not required to sample p *exactly*. We shall see in Section II-A how to sample a Bernoulli distribution with an arbitrary $p \in [0,1]$ as parameter (not the same p as our probability distribution for GHZ) using a sequence of approximants converging to p and using an expected number of only five unbiased identically independently distributed (i.i.d.) random bits. Subsequently, we shall see in Section II-B how to sample p_0 (or p_1) by modifying von Neumann's rejection algorithm in a way that it uses sequences of approximants and unbiased i.i.d. random bits. For simulating exactly the GHZ distribution, an expected number of $6n + 17$ perfect random bits is sufficient. For simplicity, we shall henceforth systematically ignore zero-probability events in our analyses.

A. Sampling a Bernoulli Distribution

Assume that only a random bit generator is available to sample a given probability distribution and that the parameters that specify this distribution are only accessible as follows: we can ask for any number of bits of each parameter, but will be charged one unit of cost per bit that is revealed. We shall also be charged for each random bit requested from the generator.

To warm up to this conundrum, consider the problem of generating a Bernoulli random variable Y with parameter $p \in [0,1]$, meaning that $Y = 0$ with probability p and $Y = 1$ otherwise. If $U = 0.U_1U_2\ldots$ is the binary expansion of a uniform $[0,1]$ random variable, i.e. $U_1, U_2, \ldots$ is our source of unbiased independent random bits, and if $p = 0.p_1p_2\ldots$ is the binary expansion of p (in case $p = 1$, we can proceed as if it were $0.p_1p_2\ldots$ with each $p_i = 1$), we compare bits U_i and p_i for $i = 1, 2, \ldots$ until for the first time $U_i \neq p_i$. Then, if $U_i = 0 < p_i = 1$, we return $Y = 0$, and if $U_i = 1 > p_i = 0$, we return $Y = 1$. It is clear that $Y = 0$ if and only if $U < p$, which happens with probability p. Therefore, Y is indeed Bernoulli(p). The expected number of bits required from p is precisely 2. The expected number of bits needed from our random bit source is also 2.

Now, suppose that the parameter p defining our Bernoulli distribution is given by $p = \cos^2(\theta/2)$, as in the case of our decomposition of the GHZ distribution. None of the parties can know θ precisely since it is distributed as a sum of θ_j's, each of which is known only by one individual party. If we could obtain as many bits in the binary expansion of p as needed (although the expected number of required bits is as little as 2), we would use the idea given above in order to sample according to this Bernoulli distribution. However, it is not possible in general to know even the first bit of p given any fixed number of bits of the θ_j's. For instance, if θ is arbitrarily close to $\pi/2$, we need arbitrarily many bits of precision about it before we can tell if the first bit in the binary expansion of $\cos^2(\theta/2)$ is 0 or 1. Nevertheless, we can use *approximations* of p, rather than *truncations*, which in turn can come from approximations (in particular truncations) of the θ_j's.

Definition 1: A k-bit approximation *of a real number x is any $\hat{x}$ such that $|x - \hat{x}| \leq 2^{-k}$. A special case of k-bit approximation is the k-bit* truncation $\hat{x} = \mathrm{sign}(x)\lfloor |x| 2^k \rfloor / 2^k$, *where* $\mathrm{sign}(x)$ *is equal to $+1$, 0 or -1 depending on the sign of x. Note that the value of k corresponds to the number of bits in the fractional part, without limitation on the size of the integer part, and that it does not take account of the sign in case it has to be transmitted too.*

We postpone to Section III-B the detail of how these approximations can be obtained in a distributed setting. For now, let us assume that we can obtain approximation $p[k]$ so that $|p[k] - p| \leq 2^{-k}$ for any desired precision k. Then, setting $U[k] = 0.U_1 \ldots U_k$ so that $U[k] \leq U < U[k] + 2^{-k}$, we have that $U \geq p$ if $U[k] \geq p[k] + 2^{-k}$ whereas $U < p$ if $U[k] + 2^{-k} \leq p[k] - 2^{-k}$, hence if $U[k] \leq p[k] - 2 \cdot 2^{-k}$. Thus, one can check if $U < p$ by generating only a finite number of bits of U and increasingly good approximations of p. These ideas are formalized in Algorithm 1. It is elementary to verify that the Y generated by this algorithm is Bernoulli(p), again because $\mathbf{P}\{U < p\} = p$ if U is a continuous uniform random variable on $[0,1]$.

The number of iterations before Algorithm 1 returns a value, which is also its required number of independent unbiased random bits, is a random variable, say K. We have seen above that $\mathbf{E}\{K\}$, the expected value of K, would be exactly 2 if we could generate arbitrarily precise truncations of p. But since we can only obtain arbitrarily precise *approximations* instead, which is why we needed Algorithm 1 in the first place, we shall have to pay the price of a small increase in $\mathbf{E}\{K\}$.

$$\begin{aligned} \mathbf{P}\{K > k\} &\leq \mathbf{P}\left\{|U[k] - p[k]| \leq \tfrac{2}{2^k}\right\} \\ &\leq \mathbf{P}\left\{|U - p| \leq \tfrac{4}{2^k}\right\} \leq \tfrac{8}{2^k}. \end{aligned}$$

Algorithm 1 Sampling a Bernoulli With Approximate p

Set $k \leftarrow 1$
Set $U[0] \leftarrow 0$
loop
  Generate an i.i.d. unbiased bit U_k
  Compute $U[k] \leftarrow U[k-1] + U_k/2^k$
  {hence $U[k] = 0.U_1 \ldots U_k$}
  Obtain $p[k]$ so that $|p[k] - p| \leq 2^{-k}$
  if $U[k] \geq p[k] + 2^{-k}$ **then**
    return $Y = 1$
  else if $U[k] \leq p[k] - 2 \cdot 2^{-k}$ **then**
    return $Y = 0$
  else
    $k \leftarrow k + 1$
  end if
end loop

Therefore,

$$\mathbf{E}\{K\} = \sum_{k=0}^{\infty} \mathbf{P}\{K > k\} \leq \sum_{k=0}^{\infty} \min\left(1, \tfrac{8}{2^k}\right) = 5.$$

B. Sampling p_0 (or p_1) in the Random Bit Model

Here, we concentrate on sampling p_0 since p_1 can be sampled in the same way, *mutatis mutandis*. Let us define

$$\left.\begin{aligned} \alpha_j &= \cos\left(\tfrac{1}{2}\left(\varphi_j - \tfrac{\pi}{2}\right)\right) = \sin\left(\tfrac{1}{2}\left(\varphi_j + \tfrac{\pi}{2}\right)\right) \quad \text{and} \\ \beta_j &= \cos\left(\tfrac{1}{2}\left(\varphi_j + \tfrac{\pi}{2}\right)\right) = -\sin\left(\tfrac{1}{2}\left(\varphi_j - \tfrac{\pi}{2}\right)\right). \end{aligned}\right\} \quad (4)$$

Clearly, $\alpha_j^2 + \beta_j^2 = 1$. Now, consider n Rademacher[2] random variables B_j that take value -1 with probability β_j^2 and $+1$ with complementary probability α_j^2. The random vector with independent components given by $(B_1, \ldots, B_n)$ is distributed according to

$$q_0(b) \stackrel{\text{def}}{=} \prod_{j \in F_b} \beta_j^2 \prod_{j \in G_b} \alpha_j^2 = \mathrm{a}_0^2(b),$$

where $F_b = \{j \mid b_j = -1\}$ and $G_b = \{j \mid b_j = +1\}$ for all $b = (b_1, \ldots, b_n) \in \{-1, +1\}^n$, and a_0 is given in (3). Similarly, the random vector with independent components given by $(-B_1, \ldots, -B_n)$ is distributed according to

$$q_1(b) \stackrel{\text{def}}{=} \prod_{j \in F_b} \alpha_j^2 \prod_{j \in G_b} \beta_j^2 = \mathrm{a}_1^2(b).$$

The key observation is that both q_0 and q_1 can be sampled without any needs for communication because each party j knows his own parameters α_j^2 and β_j^2, which is sufficient to draw independently local Rademacher random variable B_j or $-B_j$. Moreover, a single unbiased independent random bit S drawn by a designated party suffices to sample collectively from distribution $q = \frac{q_0 + q_1}{2}$, provided this bit is transmitted to all parties: everybody samples according to q_0 if $S = 0$ or to q_1 if $S = 1$. Furthermore, it follows from (2) that $p_0(b) + p_1(b) = \mathrm{a}_0^2(b) + \mathrm{a}_1^2(b) = q_0(b) + q_1(b)$ for every $b \in \{-1, +1\}^n$, and therefore $p_0(b) \leq q_0(b) + q_1(b) = 2q(b)$.

[2]A Rademacher random variable is just like a Bernoulli, except that it takes value $+1$ or -1, rather than 0 or 1.

Algorithm 2 Sampling p_0 Using von Neumann's Algorithm

repeat
  Generate independent Rademacher random variables $B_1, \ldots, B_n$ with parameters $\alpha_1^2, \ldots, \alpha_n^2$
  Generate an unbiased independent random bit S
  if $S = 1$ **then**
    set $B \leftarrow (B_1, \ldots, B_n)$
  else
    set $B \leftarrow (-B_1, \ldots, -B_n)$
  end if
  {Random variable B is now sampled according to q}
  Generate U uniformly on $[0, 1]$
until $2Uq(B) \leq p_0(B)$

The relevance of all these observations is that we can apply von Neumann's rejection algorithm [19] to sample p_0 since it is bounded by the small constant 2 that multiplies easy-to-draw probability distribution q. For the moment, we assume once again the availability of a continuous uniform random generator, which we shall later replace by a source of unbiased independent random bits. We also assume for the moment that we can compute the α_j's, $q(b)$ and $p_0(b)$ exactly. This gives rise to Algorithm 2.

By the general principle of von Neumann's rejection algorithm, probability distribution p_0 is successfully sampled after an expected number of 2 iterations round the loop because $p_0(b) \leq 2q(b)$ for all $b \in \{-1, +1\}^n$. Within one iteration, two expected independent unbiased random bits suffice to generate each of the n Rademacher random variables by a process similar to what is explained in the second paragraph of Section II-A. Hence an expected total of $2n + 1$ random bits are needed each time round the loop for an expected grand total of $4n + 2$ bits to sample p_0. But of course, this does not take account of the (apparent) need to generate the continuous uniform $[0, 1]$ random variable U. It follows that the expected total amount of work required by Algorithm 2 is $O(n)$, provided we count infinite real arithmetic at unit cost. Furthermore, the time taken by this algorithm, divided by n, is stochastically smaller than a geometric random variable with constant mean, so its tail is exponentially decreasing.

Now, we modify and adapt this algorithm to eliminate the need for the continuous uniform U (and hence its generation), which is not allowed in the random *bit* model. Furthermore, we eliminate the need for infinite real arithmetic and for the exact values of $q(B)$ and $p_0(B)$, which would be impossible to obtain in our distributed setting since the parameters needed to compute these values are scattered among all parties, and replace them with approximations—we postpone to Section III the issue of how these approximations can be computed. (On the other hand, arbitrarily precise values of the α_j's *are* available to generate independent Rademacher random variables with these parameters because each party will be individually responsible to generate his own Rademacher variable.)

Algorithm 3 Accepting Condition for Algorithm 2

1: {Note: $B \in \{-1,+1\}^n$ is given to the algorithm, generated according to $q = \frac{q_0+q_1}{2}$}
2: Set $k \leftarrow 1$
3: Set $U[0] \leftarrow 0$
4: **loop**
5: Generate an i.i.d. unbiased bit U_k
6: Compute $U[k] \leftarrow U[k-1] + U_k/2^k$ {hence $U[k] = 0.U_1 \ldots U_k$}
7: Compute L_k and R_k from B
8: **if** $U[k]\,L_k - R_k < -\frac{4}{2^k}$ **then**
9: return $Y = 1$
10: **else if** $U[k]\,L_k - R_k > \frac{4}{2^k}$ **then**
11: return $Y = 0$
12: **else**
13: $k \leftarrow k+1$
14: **end if**
15: **end loop**

In each iteration of Algorithm 2, we generate a random variable B according to distribution q (lines 2 to 8) and it appears that we need the exact values of $q(B)$ and $p_0(B)$, as well as a uniform real random variable U, in order to decide on the last line whether to accept this B or reject it and start all over again. However, testing if $2Uq(B) \le p_0(B)$ can be achieved with finite precision on all these parameters. For this purpose, we adapt the method developed for Algorithm 1 to generate a Bernoulli random variable Y equal to 1 with probability exactly equal to $p_0(B)/2q(B)$.

Again, we denote by $U[k]$ the k-bit truncation of U, so that $U[k] \le U \le U[k] + 2^{-k}$. Furthermore, we use L_k (L for *left*) and R_k (R for *right*) to denote k-bit approximations of $2q(B)$ and $p_0(B)$, respectively, so that $|L_k - 2q(B)| \le 2^{-k}$ and $|R_k - p_0(B)| \le 2^{-k}$. Then, we use ε_k to denote the real number in interval $[-1, 1]$ such that

$$\begin{aligned} |2Uq(B) - U[k]L_k| &= \left|U\left(L_k + \frac{\varepsilon_k}{2^k}\right) - U[k]L_k\right| \\ &= \left|\big(U - U[k]\big)L_k + \frac{U\varepsilon_k}{2^k}\right| \\ &\le \frac{L_k}{2^k} + \frac{U}{2^k} \;\le\; \frac{3}{2^k}. \end{aligned}$$

Furthermore, because R_k is a k-bit approximation of $p_0(B)$,

$$|R_k - p_0(B)| \le \frac{1}{2^k}.$$

Thus, we know that $Y = 1$ whenever

$$U[k]L_k + 3/2^k < R_k - 1/2^k,$$

whereas $Y = 0$ whenever

$$U[k]L_k - 3/2^k > R_k + 1/2^k.$$

Otherwise, we are in the uncertainty zone and we need more bits of U, $q(B)$ and $p_0(B)$ before we can decide on the value of Y. This is formalized in Algorithm 3.

It follows from the above discussion that this algorithm can serve to sample Bernoulli random variable Y, in effect replacing line 9 in Algorithm 2 and providing the terminating condition "**until** $Y = 1$" as its line 10. But how many iterations does Algorithm 3 require? Let K be a random variable corresponding to the value of k upon exiting from the **loop** in this algorithm, which is the number of times round the loop and hence the number of bits needed from U and the precision in $2q(B)$ and $p_0(B)$ required in order to sample exactly Bernoulli random variable Y. We are interested in an upper bound on $\mathbf{E}\{K\}$, the expected value of K.

If the algorithm has not yet halted after having processed $U[k]$, L_k and R_k, it is because $|U[k]\,L_k - R_k| \le 4/2^k$. In this case, it is elementary to verify that

$$\big|2Uq(B) - p_0(B)\big| \le \frac{8}{2^k}$$

by rewriting $2Uq(B) - p_0(B)$ as the sum

$$\begin{aligned} &2Uq(B) - U[k]L_k \\ &+ R_k - p_0(B) \\ &+ U[k]\,L_k - R_k \end{aligned}$$

and invoking the triangle inequality on the fact that the absolute value of the three terms above is upper-bounded by $3/2^k$, $1/2^k$ and $4/2^k$, respectively. Therefore, given any B, using δ_B to denote $\frac{1}{2}\frac{8}{2^k}\frac{1}{q(B)}$, we have

$$\begin{aligned} \mathbf{P}\{K > k \mid B\} &\le \mathbf{P}\{|2Uq(B) - p_0(B)| \le 8/2^k \mid B\} \\ &= \mathbf{P}\left\{U \in \left(\frac{p_0(B)}{2q(B)} - \delta_B\,,\ \frac{p_0(B)}{2q(B)} + \delta_B\right)\right\} \\ &\le 2\delta_B = \frac{8}{2^k}\frac{1}{q(B)}. \end{aligned}$$

Thus, using k_0 to denote $3 + \lceil \log_2(1/q(B))\rceil$,

$$\begin{aligned} \mathbf{E}\{K \mid B\} &= \sum_{k=0}^{\infty} \mathbf{P}\{K > k \mid B\} \\ &\le \sum_{k=0}^{\infty} \min\left(1, \frac{8}{2^k q(B)}\right) \\ &\le \sum_{k<k_0} 1 + \sum_{k\ge k_0} \frac{8}{2^k q(B)} \\ &\le 5 + \log_2\left(\frac{1}{q(B)}\right). \end{aligned}$$

The last step uses the fact that $x + 2^{1-x} \le 2$ for all $0 \le x < 1$, where $x = \lceil \log_2(1/q(B))\rceil - \log_2(1/q(B))$.

Finally, we uncondition in order to conclude:

$$\begin{aligned} \mathbf{E}\{K\} &\le 5 + \sum_{b\in\{-1,+1\}^n} q(b)\log_2\left(\frac{1}{q(b)}\right) \\ &= H(q) + 5 \\ &\le n + 5, \end{aligned} \tag{5}$$

where $H(q)$ denotes the Shannon entropy of $q = \frac{q_0+q_1}{2}$.

III. Communication Complexity of Sampling

In this section, we consider the case in which the sampler of the previous section no longer has full knowledge of the GHZ distribution to be simulated. The sampler, whom we call *the leader* in a distributed setting, has to communicate through classical channels in order to obtain partial knowledge of the parameters belonging to the other parties. Partial knowledge results in approximations of the parameters involved in sampling the GHZ distribution, but, as we saw in the previous section, we know how to sample *exactly* in the random bit model using such approximations. We consider two models of communication: in the *sequential model*, the leader has a direct channel with everyone else and all the communication has to take place sequentially because the leader cannot listen to everyone at the same time; in the *parallel model*, parties communicate with one another in a tree-structured way, with the leader at the root, which makes it possible to save on communication *time*, at the expense of a small increase in the total number of bits that need to be communicated. Unless specified otherwise, mostly in Section V-A, the sequential model is implicitly assumed.

A. Approximating Sums and Products of Bounded Numbers

We shall need to approximate sums and products of numbers for which we already have approximations or truncations.

Theorem 1: Let k and υ be integers and consider any two real numbers x and y in interval $[-2^\upsilon, 2^\upsilon]$. Let $\hat{x}$ and $\hat{y}$ be arbitrary k-bit approximations of x and y, respectively, also restricted to lie in interval $[-2^\upsilon, 2^\upsilon]$.[3] *Then,*

1) *$\hat{x}+\hat{y}$ is a $(k-1)$-bit approximation of $x+y$;*
2) *$\hat{x}/2$ is a $(k+1)$-bit approximation of $x/2$; and*
3) *$\hat{x}\hat{y}$ is a $(k-\upsilon-1)$-bit approximation of xy.*

Proof:

1) $|(\hat{x}+\hat{y})-(x+y)| = |(\hat{x}-x)+(\hat{y}-y)|$
$\le |\hat{x}-x|+|\hat{y}-y| \le 2^{-k}+2^{-k} = 2^{-(k-1)}$;
2) $|\frac{\hat{x}}{2}-\frac{x}{2}| = \frac{|\hat{x}-x|}{2} \le 2^{-k}/2 = 2^{-(k+1)}$; and
3) $|xy-\hat{x}\hat{y}| = \frac{1}{2}\big|(x+\hat{x})(y-\hat{y})+(x-\hat{x})(y+\hat{y})\big|$
$\le \frac{1}{2}\big(|(x+\hat{x})(y-\hat{y})|+|(x-\hat{x})(y+\hat{y})|\big)$
$\le \frac{1}{2}\big((|x|+|\hat{x}|)2^{-k}+2^{-k}(|y|+|\hat{y}|)\big)$
$\le \frac{1}{2}\big((2^\upsilon+2^\upsilon)2^{-k}+2^{-k}(2^\upsilon+2^\upsilon)\big) = 2^{-(k-\upsilon-1)}$,

where we used the triangle inequality $|a+b| \le |a|+|b|$. □

Corollary 1: Let k, υ, x, y, $\hat{x}$ and $\hat{y}$ be as in Theorem 1.

1) *$\hat{x}^2+\hat{y}^2$ is a t-bit approximation of x^2+y^2; and*
2) *$\frac{1}{2}(\hat{x}\pm\hat{y})^2$ is a t-bit approximation of $\frac{1}{2}(x\pm y)^2$,*

where $t = k-\upsilon-2$.

Proof: This follows from Theorem 1, using the fact that the sum of two numbers in interval $[-2^\upsilon, 2^\upsilon]$ lies in interval $[-2^{\upsilon+1}, 2^{\upsilon+1}]$ and that if $\hat{y}$ is a k-bit approximation of y, then $-\hat{y}$ is a k-bit approximation of $-y$. □

Corollary 2: Let k and $n < 2^k$ be integers and let $\{x_j\}_{j=1}^n$ and $\{\hat{x}_j\}_{j=1}^n$ be real numbers and their k-bit approximations, all in interval $[-1, 1]$. Then $\prod_{j=1}^n \hat{x}_j$ is a $(k-\lceil\log_2 n\rceil)$-bit approximation of $\prod_{j=1}^n x_j$.

[3] In case $\hat{x}$ and/or $\hat{y}$ would lie slightly outside $[-2^\upsilon, 2^\upsilon]$, they can be safely pushed back on the frontier of this interval.

Proof: Let us place the $\hat{x}_j$'s in the leaves of a binary tree of height $\lceil\log_2 n\rceil$. If each internal node represents the product of its two children, the corollary follows from repeated use of Theorem 1, using $\upsilon = 0$, since we lose one bit of precision at each level up the tree until we reach $\prod_{j=1}^n \hat{x}_j$ at the root. □

B. Sampling a Bernoulli Whose Parameter Is Distributed

In order to sample the GHZ distribution, we know from Section II that we must first sample the Bernoulli distribution with parameter $\cos^2(\theta/2)$, where $\theta = \sum_{j=1}^n \theta_j$. Let us say that the leader is party number 1. Since he knows only θ_1, he must communicate with the other parties to obtain partial knowledge about θ_j for $j \ge 2$. The problem of sampling a Bernoulli distribution with probability $\cos^2(\theta/2)$ reduces to learning the sum θ with sufficient precision in order to use Algorithm 1.

To compute a k-bit approximation of

$$\cos^2(\theta/2) = \cos^2\Big(\sum_{j=1}^n \theta_j/2\Big),$$

define $\vartheta = \theta/2$ and $\vartheta_j = \theta_j/2$ for each j. If the leader obtains an ℓ-bit approximation $\hat{\vartheta}_j$ of each ϑ_j, $j \ge 2$, and if we define $\hat{\vartheta} = \sum_{j=1}^n \hat{\vartheta}_j$, we need to find the value of ℓ for which $\cos^2(\hat{\vartheta})$ is a k-bit approximation of $\cos^2(\vartheta)$. By virtue of standard results on Taylor series expansion, we have

$$|\cos^2(\vartheta)-\cos^2(\hat{\vartheta})| \le \Big(\sup_{(\vartheta_1,\ldots,\vartheta_n)} \|\nabla\big(\cos^2(\vartheta)\big)\|\Big)\|\vartheta-\hat{\vartheta}\|$$
$$\le \sqrt{n}\,\frac{\sqrt{n}}{2^\ell} = \frac{n}{2^\ell},$$

where $\|\cdot\|$ denotes the Euclidean norm of a vector. Hence, it suffices to choose $\ell = k+\lceil\log_2 n\rceil$ in order to conclude as required that $|\cos^2(\vartheta)-\cos^2(\hat{\vartheta})| \le 2^{-k}$. Taking into account the integer part of each ϑ_j, which must also be communicated, and remembering that $0 \le \vartheta_j \le 2\pi$ since it is an angle,[4] the required number of bits communicated in the sequential model is $(n-1)(\ell+3) = (n-1)\big(k+3+\lceil\log_2 n\rceil\big)$, which is $O(kn+n\log n)$. In our case, the expected value of k is bounded by 5 (see the analysis of the Bernoulli sampling Section II-A), so that this operation requires an expected communication of $O(n\log n)$ bits.

C. Distributed Version of von Neumann's Algorithm

Once the leader has produced a bit Z according to a Bernoulli distribution with parameter $\cos^2(\theta/2)$, he still needs to sample p_Z. Of course, he does not know φ_j for $j \ge 2$. In order to apply von Neumann's rejection method from Section II-B (Algorithms 2 and 3), he needs the ability to learn with sufficient precision the products $a_0(B) = \prod_{j=1}^n c_j$ and $a_1(B) = (-1)^n \prod_{j=1}^n s_j$, in which $c_j = \cos\big(\frac{1}{2}(\varphi_j - \frac{\pi}{2}B_j)\big)$ and $s_j = \sin\big(\frac{1}{2}(\varphi_j - \frac{\pi}{2}B_j)\big)$, given that the B_j's are nonidentical independent Rademacher distributions with parameters α_j^2 or β_j^2, $1 \le j \le n$, defined in (4). Once these products

[4] Actually, $0 \le \vartheta_j \le \pi$ since ϑ_j is a *half* angle and one fewer bit is needed to communicate its integer part, but we prefer to consider here the more general case of approximating the cosine square of a sum of arbitrary angles.

are known with $k+2$ bits of precision, the left and right k-bit approximations L_k and R_k are easily computed by virtue of Corollary 1, using $\upsilon = 0$. This is the information needed at line 7 of Algorithm 3.

It follows from Corollary 2 that the leader can compute the required $(k+2)$-bit approximations of $\mathrm{a}_0(B)$ and $\mathrm{a}_1(B)$ if he obtains ℓ-bit approximations $\hat{c}_j$ and $\hat{s}_j$ of each party's c_j and s_j, where $\ell = k + 2 + \lceil \log_2 n \rceil$. *Notice that each party knows exactly his own c_j and s_j, and hence $\hat{c}_j$ and $\hat{s}_j$ can be transmitted directly to the leader, rather than approximations of the φ_j's.* These ℓ-bit approximations can in fact be ℓ-bit truncations, requiring the transmission of $\ell + 1$ bits, taking account of the sign, for a grand total of $2(n-1)(k+3+\lceil \log_2 n \rceil)$ bits that must be transmitted to the leader, which is $O(kn + n \log n)$. For our specific application of simulating the GHZ distribution, we proved at the end of Section II-B that the expected value of k is bounded by $n+5$. It follows that an expected communication cost of $O(n^2)$ bits suffices to sample the GHZ distribution, as we shall prove formally in the next section.

IV. Protocol for Sampling the GHZ Distribution

We are finally ready to glue all the pieces together into Algorithm 4, which samples exactly the GHZ distribution under arbitrary von Neumann measurements, thus solving our conundrum. Its correctness is proved below, and it is shown that the expected amount of randomness used in this process is upper-bounded by $O(n)$ bits and an expected $O(n^2)$ bits of communication suffice to complete the task. Fewer bits suffice when the measurements are equatorial or when all or almost all of them are in the computational basis or nearly so.

A. Correctness of the Protocol

In line 1, the leader samples a Bernoulli Z with parameter $\cos^2\left(\sum_{j=1}^n \theta_j/2\right)$, which allows him to decide whether to sample B according to p_0 or p_1. Notice that he does not have to inform the other parties of this decision since they do not need to know if the sampling will be done according to p_0 or p_1. In Section III-B, we showed how to sample exactly a Bernoulli with parameter $\cos^2\left(\sum_{j=1}^n \theta_j/2\right)$ even when the θ_j's are not known to the leader for $j \geq 2$.

The part within the outer **repeat** loop (lines 2 to 25) is essentially von Neumann's rejection algorithm, which has been adapted and modified to work in a distributed scenario. The leader must first decide which of q_0 or q_1 to sample. For this purpose, he generates an unbiased random bit S and broadcasts it to the other parties. Sampling either q_0 or q_1 can now be done locally and independently by each party j, yielding a tentative $B_j \in \{-1, +1\}$. The parties will output these B_j's only at the end, provided this round is not rejected. To make this decision, each party computes locally $c_j = \cos\left(\frac{1}{2}\left(\varphi_j - \frac{\pi}{2}B_j\right)\right)$ and $s_j = \sin\left(\frac{1}{2}\left(\varphi_j - \frac{\pi}{2}B_j\right)\right)$, which will be sent bit by bit to the leader upon request, thus allowing him to compute increasingly precise approximations L_k and R_k of $2q(B)$ and $p_Z(B)$, respectively. These values are used to determine whether a decision can be made to accept or reject this particular B, or whether more information is needed to make this decision. As shown at the end of Section II-B, the expected number of bits needed in L_k and R_k before we can break out of the inner **loop** (lines 11 to 24) is

Algorithm 4 Complete Protocol for GHZ Sampling

1: The leader, who is party number 1, communicates with the other parties in order to obtain increasingly precise approximations of $\theta = \sum_{j=1}^n \theta_j$ until he can sample random bit Z according to *exact* Bernoulli random distribution with parameter $\cos^2(\theta/2)$ by virtue of applying Algorithm 1
{Now entering von Neumann's rejection algorithm, adapted to our distributed setting, for sampling p_Z}
2: **repeat**
3: The leader generates a fair random bit S and broadcasts it to the other parties
{The bit S determines whether to sample q_0 or q_1}
4: Locally, each party j generates a random $B_j \in \{-1, +1\}$ according to an independent Rademacher distribution so that $B_j = +1$ with probability $\cos^2\left(\frac{1}{2}\left(\varphi_j - \frac{\pi}{2}\right)\right)$
5: **if** $S = 1$ **then**
6: Each party does $B_j \leftarrow -B_j$
7: **end if**
{Random variable $B = (B_1, \ldots, B_n)$ is sampled according to $q = \frac{q_0+q_1}{2}$}
{The leader starts talking with the other parties to decide whether or not to accept B}
8: Each party computes $c_j = \cos\left(\frac{1}{2}\left(\varphi_j - \frac{\pi}{2}B_j\right)\right)$ and $s_j = \sin\left(\frac{1}{2}\left(\varphi_j - \frac{\pi}{2}B_j\right)\right)$
9: The leader sets $k \leftarrow 1$
10: The leader sets $U[0] \leftarrow 0$
11: **loop**
12: The leader generates an i.i.d. unbiased bit U_k
13: The leader computes $U[k] \leftarrow U[k-1] + U_k/2^k$ {hence $U[k] = 0.U_1 \ldots U_k$}
14: The leader requests $(k + 2 + \lceil \log_2 n \rceil)$-bit truncations of c_j and s_j from each party $j \geq 2$
15: The leader computes $(k+2)$-bit approximations of $\mathrm{a}_0(B)$ and $\mathrm{a}_1(B)$
16: The leader computes k-bit approximations L_k of $2q(B) = \mathrm{a}_0^2(B) + \mathrm{a}_1^2(B)$ and R_k of $p_Z(B)$
17: **if** $U[k]L_k - R_k < -\frac{4}{2^k}$ **then**
18: Set $Y \leftarrow 1$ and **break from the loop**
19: **else if** $U[k]L_k - R_k > \frac{4}{2^k}$ **then**
20: Set $Y \leftarrow 0$ and **break from the loop**
21: **else**
22: Set $k \leftarrow k+1$ and **continue the loop**
{The leader does not yet have enough information to decide whether to accept or reject B; therefore, he needs more information from all the other parties in order to compute one more bit of precision on $\mathrm{a}_0(B)$ and $\mathrm{a}_1(B)$}
23: **end if**
24: **end loop**
25: **until** $Y = 1$ {accepting}
26: The leader informs the other parties that the simulation is complete and, therefore, that the time has come for each party j (including the leader himself) to output his current value of B_j

$k \leq n+5$. At that point, flag Y tells the leader whether or not this was a successful run of von Neumann's rejection algorithm. If $Y = 0$, the entire process has to be restarted from scratch, except for the initial Bernoulli sampling, at line 3. On the other hand, once the leader gets $Y = 1$, he can finally tell the other parties that they can output their B_j's because, according to von Neumann's rejection algorithm, this signals that the vector $(B_1, \ldots, B_n)$ is distributed according to p_Z, hence also according to p. Furthermore, we know from von Neumann's rejection algorithm that we have an expectation of two rounds of the outer **repeat** loop before we can thus conclude successfully.

B. Expected Randomness and Communication Cost

The expected amount of randomness used in this process is upper-bounded by $6n + 17$ bits. This is calculated as follows: the expected number of bits for sampling Bernoulli Z is bounded by 5. This is followed by an expectation of two rounds of von Neumann's rejection algorithm (the outer **repeat** loop). In each of these rounds, we need 1 bit for S and expect 2 bits for each of the B_j's (hence $1 + 2n$ in total), before entering the inner **loop**. The expected number of times round this loop is bounded by $n + 5$, and one more random bit U_k is needed each time. Putting it all together, the expected number of random bits is bounded by $5+2(1+2n+(n+5)) = 6n+17$.

The expected amount of communication is dominated by the leader's need to obtain increasingly accurate approximations of c_j and s_j from all other parties at line 14 in order to compute increasingly accurate approximations L_k and R_k, which he needs in order to decide whether or not to break from the inner **loop** and, in such case, whether or not to accept B as final output. On the k^{th} time round the loop, the leader needs $k + 2 + \lceil \log_2 n \rceil$ bits of precision plus one bit of sign about each c_j and s_j, $j \geq 2$, in addition to having full knowledge about his own c_1 and s_1. According to Section III-C, this suffices for the leader to compute $(k+2)$-bit approximations of $\mathrm{a}_0(B)$ and $\mathrm{a}_1(B)$, which in turn suffice by virtue of Corollary 1 to obtain k-bit approximations L_k of $2q(B) = q_0(B) + q_1(B) = \mathrm{a}_0^2(B) + \mathrm{a}_1^2(B)$ and R_k of $p_Z(B) = \frac{1}{2}\big(\mathrm{a}_0(b) + (-1)^Z \mathrm{a}_1(b)\big)^2$. The need for the leader to obtain from the other parties these increasingly precise approximations of c_j and s_j would be very expensive if all their bits had to be resent each time round the loop, with increasing values of k. Fortunately, this process works well because the parties actually send *truncations* of these values to the leader at line 14: each truncation simply adds one bit of precision to the previous one. Hence, it suffices for the leader to request $2(4 + \lceil \log_2 n \rceil)$ bits from each other party at the onset, when $k = 1$, and only two additional bits per party are needed afterwards for each subsequent trip round the loop (one for c_j and one for s_j). All counted, a total of $2(n-1)(k+3+\lceil \log_2 n \rceil)$ bits will have been requested from all other parties by the time we have gone through the inner **loop** k times. Since the expected value of k upon exiting this loop is bounded by $n + 5$, the expected number of bits that have to be communicated to the leader to complete von Neumann's rejection algorithm (lines 2 to 25) is bounded by $2(n-1)((n+5)+3+\lceil \log_2 n \rceil)$. This is $O(n^2)$ expected bits of communication. The additional amount of communication required to sample Bernoulli Z at line 1 (which is $(n-1)(8+\lceil \log_2 n \rceil)$ bits according to Section III-B) and for the leader to broadcast to all parties the value of S, as well as synchronization bits by which he needs to inform the other parties of success or failure each time round the main loop is negligible. All counted, Algorithm 4 needs $O(n)$ bits of randomness and $O(n^2)$ bits of communication in order to sample exactly the GHZ distribution under arbitrary von Neumann measurements.

The analysis above applies regardless of the set of von Neumann measurements that have to be simulated. In some cases, however, it is very pessimistic because the expected number of times round the inner **loop** is bounded by $\mathbf{E}\{K\} \leq H(q) + 5$ according to (5), where $H(q)$ is the entropy of distribution $q = \frac{q_0+q_1}{2}$. Until now, we had simply used the fact that $H(q) \leq n$ to conclude that $\mathbf{E}\{K\} \leq n + 5$. However, $H(q)$ can be much smaller than n for some q. In general,

$$H(q) \leq 1 + (H(q_0) + H(q_1))/2$$

and

$$H(q_0) = H(q_1) = \sum_{j=1}^{n} H_2(\alpha_j^2),$$

where H_2 is the binary entropy function and the α_j's are given in (4). In the extreme case of measurements in the computational basis, which corresponds to $\varphi_j = \pm\pi/2$ and hence $\alpha_j \in \{0, 1\}$, we have $H_2(\alpha_j^2) = 0$ for all j. It follows that $H(q) = 1$, hence the expected number of times round the inner **loop** is bounded by 6, and therefore Algorithm 4 needs only an expectation of $O(n \log n)$ bits of communication in order to sample exactly the GHZ distribution under computational-basis von Neumann measurements. Of course, $O(n)$ bits of communication would suffice, even in the worst case, if we knew ahead of time that all measurements are in the computational basis, but our protocol works seamlessly with $O(n \log n)$ expected bits of communication even if the measurements are not *exactly* in the computational basis, provided $H_2(\alpha_j^2)$ is small enough, and even if up to $O(\log n)$ of the measurements are arbitrary. The effect of such measurements on the required expected amount of randomness is less dramatic since replacing "$n + 5$" by "6" in the analysis above merely reduces the expected number of random bits from $6n + 17$ bits to $4n + 19$. In the case of measurements in the computational basis, however, the parties can generate their local Rademacher variables without any need for randomness since they become deterministic. Hence, provided we modify the protocol accordingly to take account of this special case, the total expected amount of required randomness is upper-bounded by 19 bits.

V. Variations on the Theme

We can modify Algorithm 4 in a variety of ways to improve different parameters at the expense of others. Here, we discuss four of these variations: the parallel model, bounding the number of rounds, the simulation of equatorial measurements, and the case in which only the leader has access to a source of randomness.

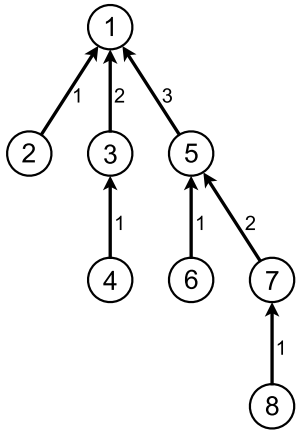

Fig. 1. Binomial tree structure defining the parallel model.

A. The Parallel Model

Until now, we have concentrated on the *sequential model* of communication, in which the leader has a direct channel with everyone else but the other participants do not communicate among themselves. This forces communication to take place sequentially because the leader cannot listen to everyone at the same time. However, as mentioned at the beginning of Section III, it is legitimate to consider a *parallel model*, in which arbitrarily many pairs of parties can communicate simultaneously. Accordingly, any number of bits can be sent and received in the same time step, provided no party has to send or receive more than one bit at any given time. This can reduce considerably the *time* required to complete our task, without entailing a significant increase in the total *number of bits* that circulate in the network.

In more detail, we have seen in Section III-C that the leader can obtain $(k+2)$-bit approximations of $\mathrm{a}_0(B) = \prod_{j=1}^{n} c_j$ and $\mathrm{a}_1(B) = (-1)^n \prod_{j=1}^{n} s_j$ at an expected communication cost of $O(n \log n + kn)$ bits. To bring this down to $O(\log^2 n + k \log n)$ *time*, we let the parties communicate with one another according to the binomial tree structure shown in Fig. 1, in which numbers in the nodes correspond to parties (the leader is number 1 at the root) and numbers next to the arrows correspond to the order in which data is transmitted. For simplicity, we may assume that n is a power of 2. To understand the algorithm, think of a tree with nodes containing s_j and c_j, for $1 \le j \le n$. Node number j can be thought of as belonging to party number j, who knows s_j and c_j exactly. We pair parties by groups of two. For a given pair, say $(j, j+1)$, with j odd, party $j+1$ sends $\hat{s}_{j+1}$ and $\hat{c}_{j+1}$ to party j, who computes $\hat{s}_j\hat{s}_{j+1}$ and $\hat{c}_j\hat{c}_{j+1}$. Then, party j is matched with party $j+2$, where $j-1$ is divisible by 4. This process gives rise to the new pair $(j, j+2)$, from which emerges the products $\hat{s}_j\hat{s}_{j+1}\hat{s}_{j+2}\hat{s}_{j+3}$ and $\hat{c}_j\hat{c}_{j+1}\hat{c}_{j+2}\hat{c}_{j+3}$, and so on up to the leader, who is party 1 at the root of the tree.

This approach is formalized in Algorithm 5, in which new variables $\tilde{c}_j$ and $\tilde{s}_j$ are introduced to hold approximations of products of increasingly many c's and s's as the process unfolds. The issue of the required precision at each level of the process has to be reconsidered because the leader will no longer receive the entire list of $\hat{c}_j$'s and $\hat{s}_j$'s since subproducts are calculated *en route* by intermediate parties, which must be truncated for transmission. For simplicity, we proceed as if all the c_j's and s_j's were nonnegative and we percolate the signs up the tree separately. We know from Theorem 1, using $v = 0$, that if $\hat{x}$ and $\hat{y}$ are ℓ-bit approximations (in particular ℓ-bit truncations) of x and y, respectively, for arbitrary real numbers x and y in $[0, 1]$ and integer ℓ, then $\hat{x}\hat{y}$ is an $(\ell-1)$-bit approximation of xy. However, $\hat{x}\hat{y}$ is *not* in general an $(\ell-1)$-bit *truncation* of xy because it could have up to 2ℓ bits of precision and we do not want to transmit so many bits up the binomial tree. There is an apparent problem if we transmit the ℓ-bit truncation of $\hat{x}\hat{y}$ instead, as we do indeed in Algorithm 5, because *it* is an $(\ell-2)$-bit approximation of xy but not necessarily an $(\ell-1)$-bit approximation. Nevertheless, it is shown in Appendix B that the recursive application of pairwise multiplications followed by truncation to ℓ bits results in the loss of only one bit of precision per subsequent level. Thus, on the i^{th} time round the **repeat** loop, $1 \le i \le \lceil \log_2 n \rceil$, the numbers calculated at line 10, even after truncation to ℓ bits, are $(\ell - i - 1)$-bit approximations of the exact product that they represent (again, up to sign, which is handled separately). It follows that the final numbers computed by the leader when $i = \lceil \log_2 n \rceil$ are $(k+2)$-bit approximations of the products of all the c_j's and the s_j's, as required, provided we start with $\ell = k + 3 + \lceil \log_2 n \rceil$.

To analyse the communication complexity of this strategy, let us consider that bits transmitted in parallel between disjoint pairs of parties count as a single time step in the global communication process. The **repeat** loop is carried out $\lceil \log_2 n \rceil$ times. Each time round this loop, parties transmit in parallel two ℓ-bit approximations, which requires $\ell + 1$ bits of communication per active party since signs must also be transmitted. It follows immediately that the parallel complexity of

Algorithm 5 Computing Products in Parallel

1: If n is not a power of 2, add virtual parties with $c_j = s_j = 1$ for $n < j \le 2^{\lceil \log_2 n \rceil}$
{These parties, being dummy, have no effect on the overall communication complexity}
2: $\ell \leftarrow k + 3 + \lceil \log_2 n \rceil$
3: **for** $j \leftarrow 1$ **to** n **in parallel do**
4: Party j does $\tilde{c}_j \leftarrow \hat{c}_j$ and $\tilde{s}_j \leftarrow \hat{s}_j$, which are ℓ-bit truncations of c_j and s_j, respectively.
5: **end for**
6: $m \leftarrow 1$
7: **repeat**
8: **for** $j \leftarrow 1$ **to** n **by step of** $2m$ **in parallel do**
9: Party $j+m$ sends $\tilde{c}_{j+m}$ and $\tilde{s}_{j+m}$ to party j
10: Party j computes $\tilde{c}_j \leftarrow \tilde{c}_j\tilde{c}_{j+m}$ and $\tilde{s}_j \leftarrow \tilde{s}_j\tilde{s}_{j+m}$, both truncated to ℓ bits
11: **end for**
12: $m \leftarrow 2m$
13: **until** $m \ge n$
14: Party 1 (the leader) outputs $\tilde{c}_1$ and $(-1)^n\tilde{s}_1$

Algorithm 5 is $2(\ell+1)\lceil \log_2 n\rceil = 2(k+4+\lceil \log_2 n\rceil)\lceil \log_2 n\rceil$, which is $O(\log^2 n + k \log n)$. Therefore, this takes $O(k \log n)$ time provided $k > \log_2 n$.

Now, let us use Algorithm 5 to replace lines 14 and 15 in Algorithm 4. This allows the leader to obtain his required $(k+2)$-bit approximations of $a_0(B)$ and $a_1(B)$ with no need for him to learn all the $(k+2+\lceil \log_2 n\rceil)$-bit truncations of c_j and s_j from each party $j \geq 2$. We have just seen that $O(k \log n + \log^2 n)$ parallel time suffices for this task. Unfortunately, this improvement is incompatible with the idea of transmitting only one more bit of information for each c_j and s_j when k is increased by 1, which was crucial in the efficiency of the sequential version of Algorithm 4 studied in Section IV. The problem stems from the fact that the ℓ-bit truncation of the product of the ℓ-bit truncations of x and y can be entirely different from the $(\ell+1)$-bit truncation of the product of the $(\ell+1)$-bit truncations of the same numbers. This is illustrated with $x = 0.1111\ldots$ and $y = 0.1001\ldots$ (in binary, of course). If we take $\ell = 3$, the truncations of x and y are 0.111 and 0.100, respectively, whose product is 0.011100. In contrast, with $\ell = 4$, the truncations of x and y are 0.1111 and 0.1001, respectively, whose product is 0.10000111. We see that the 3-bit truncation of the product of the 3-bit truncations is 0.011, whereas the 4-bit truncation of the product of the 4-bit truncations is 0.1000, which are different on each and every bit of the fractional part! This demonstrates the fact that the bits going up the binomial tree in Algorithm 5 can change drastically from one run to the next even if a single bit of precision is added to all nodes at the bottom level, and therefore that we have to start afresh for each new value of k. As a consequence, the use of Algorithm 5 to replace lines 14 and 15 in Algorithm 4 results in an "improvement" in which we expect to have to transmit $\Omega(n^3)$ bits, taking $\Omega(n^2 \log n)$ parallel time to do so!

Fortunately, there is an easy cure to this problem, which we only sketch here. In addition to using Algorithm 5 to replace lines 14 and 15 in Algorithm 4, we also change line 22 from "$k \leftarrow k+1$" to "$k \leftarrow 2k$". Even though parties have to transmit up the binomial tree the entire $(k+3+\lceil \log_2 n\rceil)$-bit truncations of each c_j and s_j for each new value of k, the work done each time round the loop is roughly equivalent to the sum of all the work done until then. Since we expect to succeed when k is roughly equal to n, the expected total parallel time is about twice $O(k \log n + \log^2 n)$ with $k \approx n$, which is simply $O(n \log n)$. The expected total number of bits communicated with this approach is slightly greater than with Algorithm 4, but remains $O(n^2)$.

B. Reducing the Number of Rounds

Algorithm 4 is efficient in terms of the number of bits of randomness as well as the number of bits of communication, but it requires an expected $O(n)$ rounds, in which the leader and all other parties take turn at sending messages. An approach to reduce the number of rounds should be rather obvious at this point, and we leave the details to the reader. If we change line 22 from "$k \leftarrow k+1$" to "$k \leftarrow 2k$", the expected number of rounds is decreased from $O(n)$ to $O(\log n)$. If in addition we start with "$k \leftarrow n$" instead of "$k \leftarrow 1$" at line 9, the expected number of rounds becomes a constant. (Alternatively, we could start with "$k \leftarrow n$" at line 9 and step with "$k \leftarrow k+n$" at line 22.)

C. Equatorial Measurements

Recall that equatorial measurements are those for which $\varphi_j = 0$ for each party j. In this case, the leader can sample according to p_0 or p_1, without any help or communication from the other parties, since he has complete knowledge of their vanished elevation angles. Therefore, he can run lines 2 to 25 of Algorithm 4 all by himself! However, he still needs to communicate in line 1 of Algorithm 4 in order to know from which of p_0 or p_1 to sample. The only remaining need for communication occurs in line 26, which has to be modified from "The leader informs the other parties that the simulation is complete" to "The leader informs the other parties of which value of $B_j \in \{-1, +1\}$ he has chosen for them".

Only line 1 requires significant communication since the new line 26 needs only the transmission of $n-1$ bits. We have already seen at the end of Section III-B that line 1, which is a distributed version of Algorithm 1, requires an expected communication of $O(n \log n)$ bits in the sequential model. This is therefore the complexity of our simulation, which is an improvement over the previously best technique known to simulate the GHZ distribution under arbitrary equatorial von Neumann measurements [14], which required an expectation of $O(n^2)$ bits of communication.

A more elegant protocol can be obtained if we use (6), which is found at the end of Appendix A: this is a simplified formula for $p(b)$ in the case of equatorial measurements. Each party j other than the leader can simply generate an independent unbiased Rademacher variable $b_j \in \{-1, +1\}$ as final output, without any consideration of his own input θ_j nor communication with anyone else, and inform the leader of this choice. It simply remains for the leader to choose his own b_1 in order to make $\prod_{j=1}^{n} b_j$ equal to $+1$ with probability $\cos^2(\theta/2)$ or -1 with complementary probability $\sin^2(\theta/2)$. For this, we still need line 1 from Algorithm 4, which requires an expected communication of $O(n \log n)$ bits.

To adapt this latter protocol to the parallel model, note that the leader does not need to know all the b_j's chosen by the other parties since he only needs their product, which is either $+1$ or -1. It is elementary to adapt Algorithm 5 in order to percolate this information to the leader up the binomial tree, at a communication cost of $O(n)$ bits but only $O(\log n)$ parallel time. One can also adapt Algorithm 5 to work with sums instead of products, which is the relevant operation to parallelize the distributed version of Algorithm 1. Sums and products are similar since if $\hat{x}$ and $\hat{y}$ are t-bit approximations of x and y, respectively, for an arbitrary integer t, then $\hat{x}+\hat{y}$ and $\hat{x}\hat{y}$ are $(t-1)$-bit approximations of $x+y$ and xy, respectively, according to Theorem 1, using $v = 0$. However, parallelizing sums is easier than products because the exact sum $\hat{x}+\hat{y}$ can be transmitted with no more bits of precision than each of $\hat{x}$ and $\hat{y}$, even though one additional bit is required

to transmit the integer part of the sum,[5] whereas the product $\hat{x}\hat{y}$ could entail twice as many bits of precision than each of $\hat{x}$ and $\hat{y}$ (this is why we needed Appendix B). In round k of the **loop** in Algorithm 1, the leader needs to obtain a k-bit approximation of $\cos^2\left(\sum_{j=1}^{n}\theta_j/2\right)$, which in turn requires the addition of $\ell = k + 3 + \lceil \log_2 n \rceil$ bits from each of the $n-1$ half-angles $\theta_j/2$ owned by the various parties $j \geq 2$. The binomial tree construction makes it possible to percolate this sum up to the leader through $\lceil \log_2 n \rceil$ levels in which it is sufficient to transmit $\ell + i$ bits up the tree for each partial sum (or initial half angle) at distance i from the leaves. The expected cost before the leader obtains the required k-bit approximation of $\cos^2\left(\sum_{j=1}^{n}\theta_j/2\right)$ is therefore $O((k+\log n)\, n)$ bits of communications but only $O((k+\log n)\log n)$ parallel time. Using once again the fact that the expected number k of rounds in Algorithm 1 is bounded by 5, the required Bernoulli variable with parameter $\cos^2\left(\sum_{j=1}^{n}\theta_j/2\right)$ can be sampled exactly after an expected communication cost of $O(n \log n)$ bits, as in the sequential model, but only $O(\log^2 n)$ parallel time. This dominates the cost of the parallel implementation of our algorithm in the case of equatorial measurements.

Note that all this information is sent up the binomial tree towards the leader. The only information that the leader has to send back down to the other parties, each time round the **loop**, serves to notify them of whether or not a more precise approximation of their azimuthal angles is required in order to complete the Bernoulli sampling. This bit can be sent down the binomial tree at the cost of $O(\log n)$ time if we reverse the arrows in Figure 1 and reorder the transmissions from the edge marked $\lceil \log_2 n \rceil$ (which is $\lceil \log_2 8 \rceil = 3$ in the figure) down to the edges marked 1.

If we consider a nonstandard model in which we only care about what happens until all parties have produced their output, we can modify the above protocol to require only one-way communication on each of the links, namely up the binomial tree, with no increase (in fact a small decrease) in expected communication and time complexities before the final output has been produced. For this, we simply remove the leader's notification to all other parties of whether or not the simulation has been completed. This means that all parties will indefinitely continue to provide the leader (who will pay no attention!) with increasingly precise approximations of the sum of their azimuthal angles, but this useless activity will take place after all parties have produced their output. Indeed, all parties other than the leader can output their randomly selected $+1$ or -1 at the very beginning of the protocol, and the leader can output his answer as soon as *he* knows that the Bernoulli sampling (Algorithm 1) has been successful.

Of course, we could have parallelized line 1 of Algorithm 4 even in the case of non-equatorial measurements. However, this would not have impacted significantly on the overall time complexity of our general solution, which remains $O(n \log n)$.

[5]One may be tempted to prevent the accumulation of large angles by reducing each sum modulo 2π before transmission up the binomial tree. However, this would void the advantage we had reaped from the fact that the fractional part of the sum of t-bit truncations (as opposed to their product) contains only t bits of precision.

D. Only the Leader Needs to Be Probabilistic

It is easy to modify almost all our protocols to require randomness only from the leader, all other parties being purely deterministic. For this, notice that the total expected amount of randomness is only $O(n)$, which is negligible compared to the total number of bits that have to be communicated. Hence, each time one party needs a random bit, he can ask the leader to provide it. This will only increase the communication cost by an expected $O(n)$ bits, which has no effect on the overall asymptotic communication complexity of our protocols. The same remark applies to the time required by our protocols in the parallel model, with the exception of the case of equatorial measurements, in which an $O(\log^2 n)$ expected time requires all parties other than the leader to choose their independent unbiased random output in parallel.

VI. Conclusion, Discussion and Open Problems

We have addressed the problem of simulating the effect of arbitrary independent von Neumann measurements on the qubits forming the general GHZ state $\frac{1}{\sqrt{2}}|0^n\rangle + \frac{1}{\sqrt{2}}|1^n\rangle$ distributed among n parties. Rather than doing the actual quantum measurements, the parties must sample the exact GHZ probability distribution by purely classical means, which necessarily requires communication in view of Bell's theorem. Our main objective was to find a protocol that solves this conundrum with a finite amount of expected communication, which had only been known previously to be possible when the von Neumann measurements are restricted to being equatorial (a severe limitation indeed). Our solution needs only an expectation of $O(n^2)$ bits of communication, which can be dispatched in $O(n \log n)$ expected time if bits can be sent in parallel according to a realistic scenario in which nobody has to send or receive more than one bit in any given step. We also improved on the former art in the case of equatorial measurements, with expectations of $O(n \log n)$ bits of communication and $O(\log^2 n)$ parallel time.

Knuth and Yao [15] initiated the study of the complexity of generating random integers (or bit strings) with a given probability distribution p, assuming only the availability of a source of unbiased identically independently distributed random bits. They showed that any sampling algorithm must use an expected number of bits at least equal to the entropy $\sum_b p(b)\log_2(1/p(b))$ of the distribution, and that the best algorithm does not need more than two additional bits. For further results on the bit model in random variate generation, see Ref. [20, Ch. XIV] and Ref. [21].

The GHZ distribution has an entropy no larger than n, and therefore Knuth and Yao have shown that it could be sampled with no more than $n+2$ expected random bits if all the parameters were concentrated in a single place. Even though we have studied the problem of sampling this distribution in a setting in which the defining parameters (here the description of the von Neumann measurements) are distributed among n parties, and despite the fact that our main purpose was to minimize communication between these parties, we were able to succeed with $6n+17$ expected random bits, which is just above six times the bound of Knuth and Yao. The amount of randomness required by our protocols does not depend

significantly on the actual measurements they have to simulate, as discussed at the end of Section IV-B. However, some sets of measurements entail a probability distribution p whose entropy $H(p)$ is much smaller than n. In the extreme case of having all measurements in the computational basis, $H(p)$ is a single bit! Can there be protocols that succeed with as few as $H(p)+2$ expected random bits, thus meeting the bound of Knuth and Yao, or failing this as few as $O(H(p))$ expected random bits, no matter how small $H(p)$ is for the given set of von Neumann measurements? Notice that all the protocols presented here require $\Omega(n)$ random bits since they ask each party to sample independently at least once a Rademacher distribution, a hurdle that can only be alleviated in the case of measurements in the computational basis. It *may* be that this problem can be solved if we put the leader in charge of drawing *all* the Rademacher variables in a single batch. But what would be the cost in terms of communication from the other parties, who will need to send sufficiently precise approximations of their elevation angles φ_j to the leader, rather than the much easier task of generating their own Rademacher variables locally?

Are our protocols optimal in terms of the amount of communication that is required? Could we simulate arbitrary von Neumann measurements as efficiently as in the case of equatorial measurements, i.e. with $O(n\log n)$ expected bits of communication? Or better? We leave this as an open question, but point out that Broadbent, Chouha and Tapp have proved an $\Omega(n\log n)$ lower bound on the *worst case* communication complexity of simulating measurements on n-partite GHZ states [22], a result that holds even for equatorial measurements, and even under the promise that $\cos\sum_{j=1}^{n}\theta_j=\pm 1$ [23]. However, this lower bound says nothing about the *expected case* complexity of these simulations, which is the subject matter studied here, and indeed we are not aware of any relevant nontrivial lower bounds. Of course, *some* communication is required with nonzero probability, in view of Bell's theorem, but this argument does not even rule out the possibility of an expectation smaller than one bit of communication.

As a recent development, which will be the subject of a follow-up paper, we have discovered how to simulate arbitrary generalized measurements (POVMs) on arbitrary n-partite states of arbitrary dimension [24]. Even though our general solution has finite bounded expected communication provided the POVMs have a finite number of possible outcomes, it is exponentially expensive in the number of participants. The method presented in the current paper is therefore much more efficient, because it uses the specific structure inherent to the case of von Neumann measurements on the generalized GHZ state. It remains an open question to find an efficient solution to the general problem, or to prove that no such solution can exist. Nevertheless, the intermediate case of simulating von Neumann measurements on an arbitrary n-partite state, in which each participant is given a single qubit, has an efficient solution [26]. This includes the case of the tripartite W state $\frac{1}{\sqrt{3}}|100\rangle+\frac{1}{\sqrt{3}}|010\rangle+\frac{1}{\sqrt{3}}|001\rangle$ and its n-partite generalization [25]. Although this entails a simulation process that is rather more complicated than what is presented here, its expected communication cost remains quadratic in the number of participants. However, we do not know how to proceed with as few random bits as we need here to simulate the GHZ state.

We leave for further research the problem of simulating multipartite entanglement with *worst-case* bounded classical communication. Clearly, an approach different from ours would be needed for this task since ours is based on von Neumann's rejection method, which intrinsically has nonzero probability to reject an arbitrarily large number of false starts. Nevertheless, there is hope that this problem might have a solution since a simulation with worst-case bounded communication is known for the case of arbitrary von Neumann measurements on Bell states [4], [6] and of equatorial von Neumann measurements on the tripartite GHZ state [13].

Appendix A
Convex Decomposition of the GHZ Distribution

Our exact simulation of the GHZ distribution hinges upon its decomposition into a convex combination of two subdistributions, which is stated at the beginning of Section II. Let us repeat (1) for convenience:

$$p(b)=\cos^2\left(\tfrac{\theta}{2}\right)p_0(b)+\sin^2\left(\tfrac{\theta}{2}\right)p_1(b),$$

where the coefficients $\cos^2(\theta/2)$ and $\sin^2(\theta/2)$ depend only on the azimuthal angles of the measurements, whereas the subdistributions p_0 and p_1 depend only on their elevation angles. This decomposition was obtained by one of us [17], [18], albeit in the usual computer science language in which von Neumann measurements are presented as a unitary transformation followed by a measurement in the computational basis. For completeness, here we derive this decomposition directly in the language of von Neumann measurements.

First, let us recall some facts, including some already mentioned in Section II. We begin with a 2×2 von Neumann measurement, which can be represented by operator

$$\begin{aligned}M&=x\sigma_1+y\sigma_2+z\sigma_3\\&=x\begin{pmatrix}0&1\\1&0\end{pmatrix}+y\begin{pmatrix}0&-\imath\\\imath&0\end{pmatrix}+z\begin{pmatrix}1&0\\0&-1\end{pmatrix}\\&=\begin{pmatrix}z&x-\imath y\\x+\imath y&-z\end{pmatrix},\end{aligned}$$

where $x^2+y^2+z^2=1$. Thus, using spherical coordinates $(\theta,\varphi)\in[0,2\pi)\times[-\pi/2,\pi/2]$, the parameters (x,y,z) can be written as

$$\begin{aligned}x&=\cos\theta\cos\varphi\\y&=\sin\theta\cos\varphi\\z&=\sin\varphi\end{aligned}$$

so that

$$M=\begin{pmatrix}\sin\varphi&e^{-\imath\theta}\cos\varphi\\e^{\imath\theta}\cos\varphi&-\sin\varphi\end{pmatrix}.$$

The spectra (set of eigenvalues) of M is $\{-1,+1\}$ and the unitary operator U that diagonalizes M is given by

$$U=\begin{pmatrix}\alpha&-\bar{\beta}\\-\beta&-\bar{\alpha}\end{pmatrix}$$

with

$$\alpha = \cos\left(\frac{\varphi}{2} - \frac{\pi}{4}\right) \text{ and } \beta = e^{\imath\theta}\sin\left(\frac{\varphi}{2} - \frac{\pi}{4}\right).$$

In other words, we have

$$M = U\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}U^\dagger.$$

The density matrix $\rho = |\Psi_n\rangle\langle\Psi_n|$ representing the GHZ state can be decomposed as

$$\frac{1}{2}\left(\bigotimes_{j=1}^n\begin{pmatrix}1&0\\0&0\end{pmatrix} + \bigotimes_{j=1}^n\begin{pmatrix}0&1\\0&0\end{pmatrix} + \bigotimes_{j=1}^n\begin{pmatrix}0&0\\1&0\end{pmatrix} + \bigotimes_{j=1}^n\begin{pmatrix}0&0\\0&1\end{pmatrix}\right).$$

Before analysing the joint probability function, note that

$$|b\rangle\langle b| = \begin{pmatrix}\delta_{+1}(b) & 0\\ 0 & \delta_{-1}(b)\end{pmatrix},$$

where $b \in \{-1,+1\}$, $|+1\rangle = \binom{1}{0}$, $|-1\rangle = \binom{0}{1}$ and δ is the Kronecker delta function $\delta_x(y) = 1$ if $x = y$ and $\delta_x(y) = 0$ if $x \neq y$. We also invite the reader to verify that

$$\begin{aligned}\mathrm{Tr}\left(\Big(|b\rangle\langle b|\Big)U\begin{pmatrix}1&0\\0&0\end{pmatrix}U^\dagger\right) &= |\alpha|^2\delta_{+1}(b) + |\beta|^2\delta_{-1}(b)\\ &= \cos^2\left(\frac{1}{2}\Big(\varphi - \frac{\pi}{2}b\Big)\right)\\ \mathrm{Tr}\left(\Big(|b\rangle\langle b|\Big)U\begin{pmatrix}0&1\\0&0\end{pmatrix}U^\dagger\right) &= -\alpha\beta\delta_{+1}(b) + \alpha\beta\delta_{-1}(b)\\ &= -e^{\imath\theta}\sin\left(\frac{1}{2}\Big(\varphi - \frac{\pi}{2}b\Big)\right)\cos\left(\frac{1}{2}\Big(\varphi - \frac{\pi}{2}b\Big)\right)\\ \mathrm{Tr}\left(\Big(|b\rangle\langle b|\Big)U\begin{pmatrix}0&0\\1&0\end{pmatrix}U^\dagger\right) &= -\bar\alpha\bar\beta\delta_{+1}(b) + \bar\alpha\bar\beta\delta_{-1}(b)\\ &= -e^{-\imath\theta}\sin\left(\frac{1}{2}\Big(\varphi - \frac{\pi}{2}b\Big)\right)\cos\left(\frac{1}{2}\Big(\varphi - \frac{\pi}{2}b\Big)\right)\end{aligned}$$

and

$$\begin{aligned}\mathrm{Tr}\left(\Big(|b\rangle\langle b|\Big)U\begin{pmatrix}0&0\\0&1\end{pmatrix}U^\dagger\right) &= |\beta|^2\delta_{+1}(b) + |\alpha|^2\delta_{-1}(b)\\ &= \sin^2\left(\frac{1}{2}\Big(\varphi - \frac{\pi}{2}b\Big)\right).\end{aligned}$$

For convenience, let

$$E_1 = \begin{pmatrix}1&0\\0&0\end{pmatrix},\quad E_2 = \begin{pmatrix}0&1\\0&0\end{pmatrix},\quad E_3 = \begin{pmatrix}0&0\\1&0\end{pmatrix} \text{ and } E_4 = \begin{pmatrix}0&0\\0&1\end{pmatrix}.$$

Given n von Neumann measurements M_j, $1 \le j \le n$, the joint probability $p(b)$ of obtaining $b \in \{-1,+1\}^n$ as a result of applying these measurements on an n-partite GHZ state is

$$\begin{aligned}p(b) &= |\langle b|U|\Psi_n\rangle|^2\\ &= \mathrm{Tr}\Big(\langle b|U|\Psi_n\rangle\big(\langle b|U|\Psi_n\rangle\big)^\dagger\Big)\\ &= \mathrm{Tr}\Big(\langle b|U|\Psi_n\rangle\langle\Psi_n|U^\dagger|b\rangle\Big)\\ &= \mathrm{Tr}\Big(\Big(|b\rangle\langle b|\Big)U\rho\,U^\dagger\Big)\\ &= \frac{1}{2}\mathrm{Tr}\Big(\Big(\bigotimes_{j=1}^n|b_j\rangle\langle b_j|\Big)\sum_{i=1}^4\bigotimes_{j=1}^n U_jE_iU_j^\dagger\Big)\\ &= \frac{1}{2}\sum_{i=1}^4\prod_{j=1}^n\mathrm{Tr}\Big(\begin{pmatrix}\delta_{+1}(b_j)&0\\0&\delta_{-1}(b_j)\end{pmatrix}U_jE_iU_j^\dagger\Big).\end{aligned}$$

For $1 \le i \le 4$, let us define

$$f_i \stackrel{\text{def}}{=} \prod_{j=1}^n\mathrm{Tr}\Big(\begin{pmatrix}\delta_{+1}(b_j)&0\\0&\delta_{-1}(b_j)\end{pmatrix}U_jE_iU_j^\dagger\Big)$$

so that $p(b) = \frac{1}{2}(f_1 + f_2 + f_3 + f_4)$. Putting these equations together, we have:

$$\begin{aligned}f_1 &= \prod_{j=1}^n\cos^2\left(\frac{1}{2}\Big(\varphi_j - \frac{\pi}{2}b_j\Big)\right)\\ f_2 &= \prod_{j=1}^n -e^{\imath\theta_j}\sin\left(\frac{1}{2}\Big(\varphi_j - \frac{\pi}{2}b_j\Big)\right)\cos\left(\frac{1}{2}\Big(\varphi_j - \frac{\pi}{2}b_j\Big)\right)\\ &= e^{\imath\theta}\prod_{j=1}^n -\sin\left(\frac{1}{2}\Big(\varphi_j - \frac{\pi}{2}b_j\Big)\right)\cos\left(\frac{1}{2}\Big(\varphi_j - \frac{\pi}{2}b_j\Big)\right)\\ f_3 &= \prod_{j=1}^n -e^{-\imath\theta_j}\sin\left(\frac{1}{2}\Big(\varphi_j - \frac{\pi}{2}b_j\Big)\right)\cos\left(\frac{1}{2}\Big(\varphi_j - \frac{\pi}{2}b_j\Big)\right)\\ &= e^{-\imath\theta}\prod_{j=1}^n -\sin\left(\frac{1}{2}\Big(\varphi_j - \frac{\pi}{2}b_j\Big)\right)\cos\left(\frac{1}{2}\Big(\varphi_j - \frac{\pi}{2}b_j\Big)\right)\\ f_4 &= \prod_{j=1}^n\sin^2\left(\frac{1}{2}\Big(\varphi_j - \frac{\pi}{2}b_j\Big)\right),\end{aligned}$$

where $\theta = \sum_{j=1}^n\theta_j$. Keeping in mind that $f_2 + f_3$ is equal to

$$2\cos\theta\prod_{j=1}^n -\sin\left(\frac{1}{2}\Big(\varphi_j - \frac{\pi}{2}b_j\Big)\right)\cos\left(\frac{1}{2}\Big(\varphi_j - \frac{\pi}{2}b_j\Big)\right)$$

and

$$\begin{aligned}x^2 + y^2 + 2xy\cos\gamma \;=\; &(x+y)^2\cos^2(\gamma/2)\\ &+ (x-y)^2\sin^2(\gamma/2)\end{aligned}$$

for all real numbers x and y, and angle γ, it suffices to take

$$\begin{aligned}x = \mathrm{a}_0(b) &= \prod_{j=1}^n\cos\left(\tfrac{1}{2}(\varphi_j - \tfrac{\pi}{2}b_j)\right)\\ y = \mathrm{a}_1(b) &= \prod_{j=1}^n -\sin\left(\tfrac{1}{2}(\varphi_j - \tfrac{\pi}{2}b_j)\right)\\ &= (-1)^n\prod_{j=1}^n\sin\left(\tfrac{1}{2}(\varphi_j - \tfrac{\pi}{2}b_j)\right)\\ \text{and}\quad \gamma &= \theta\end{aligned}$$

to conclude that

$$p(b) = \cos^2\left(\tfrac{\theta}{2}\right)p_0(b) + \sin^2\left(\tfrac{\theta}{2}\right)p_1(b)$$

where

$$p_0(b) = \tfrac{1}{2}\big(\mathrm{a}_0(b) + \mathrm{a}_1(b)\big)^2 \quad \text{and}$$
$$p_1(b) = \tfrac{1}{2}\big(\mathrm{a}_0(b) - \mathrm{a}_1(b)\big)^2.$$

This is precisely the convex decomposition for $p(b)$ that was claimed at the beginning of Section II.

As a "reality check", we analyse this formula for the special case of equatorial measurements, in which all elevation angles vanish. The formulas for $\mathrm{a}_0(b)$ and $\mathrm{a}_1(b)$, and therefore those for $p_0(b)$ and $p_1(b)$, become very simple when $\varphi_j = 0$ for all j. For any $b = (b_1, \ldots, b_n) \in \{-1, +1\}^n$, let us define

$$s(b) = \prod_{j=1}^{n} b_j \in \{-1, +1\} \quad \text{and}$$
$$X = \{b \in \{-1, +1\}^n \mid s(b) = +1\}.$$

We see easily that $\mathrm{a}_0(b) = 2^{-n/2}$ and $\mathrm{a}_1(b) = (-1)^{s(b)}2^{-n/2}$. Therefore

$$p_0(b) = \begin{cases} 2^{1-n} & \text{if } b \in X \\ 0 & \text{if } b \notin X \end{cases}$$

and

$$p_1(b) = \begin{cases} 0 & \text{if } b \in X \\ 2^{1-n} & \text{if } b \notin X. \end{cases}$$

Hence,

$$p(b) = \begin{cases} 2^{1-n}\cos^2\big(\frac{\theta}{2}\big) & \text{if } b \in X \\ 2^{1-n}\sin^2\big(\frac{\theta}{2}\big) & \text{if } b \notin X. \end{cases} \tag{6}$$

Thus, in the case of equatorial measurements, we obtain a uniformly distributed $b \in X$ with probability $\cos^2(\theta/2)$ or a uniformly distributed $b \in \{-1, +1\}^n \setminus X$ with complementary probability $\sin^2(\theta/2)$. From this, it follows immediately that the expected value of the product of the b_j's is equal to the cosine of the sum of the azimuthal angles because

$$\mathbf{E}\Big\{\prod_{j=1}^{n} b_j\Big\} = \cos^2\big(\tfrac{\theta}{2}\big) \times (+1) + \sin^2\big(\tfrac{\theta}{2}\big) \times (-1)$$
$$= \cos^2\big(\tfrac{\theta}{2}\big) - \sin^2\big(\tfrac{\theta}{2}\big) = \cos\theta = \cos\Big(\sum_{j=1}^{n}\theta_j\Big).$$

It follows equally easily that $\mathbf{E}\big\{\prod_{j\in J} b_j\big\} = 0$ for any nonempty $J \subsetneq \{1, \ldots, n\}$, and therefore all the marginal probability distributions obtained by tracing out one or more of the parties are uniform. Those well-known facts were indeed the formulas used in the prior art of simulating equatorial measurements on GHZ states [12]–[14].

Appendix B
Approximations and Truncations of Products

In this appendix, we restrict our attention to the multiplication of real numbers in the interval [0, 1] because this is what is relevant to the analysis of the parallel model, in which we need to approximate the product of sines and cosines sent up the binomial tree of Figure 1. It is sufficient, again for simplicity, to concentrate on positive numbers because the signs can be percolated independently up the binomial tree.

Consider any $t \in [0, 1]$ and positive integer ℓ. Recall from Definition 1 that the ℓ-bit truncation of t is $\lfloor t2^\ell \rfloor / 2^\ell$ because t is nonnegative. This ℓ-bit truncation is obviously an ℓ-bit approximation as well:

$$\left| \frac{\lfloor t2^\ell \rfloor}{2^\ell} - t \right| \le \frac{1}{2^\ell}.$$

Suppose now that we have two numbers x_j and y_j at level j in the binomial tree inherent to Algorithm 5, such that both numbers lie in interval [0, 1]. We can express x_j and y_j recursively using the numbers $x_{j-1,1}$, $x_{j-1,2}$, $y_{j-1,1}$, and $y_{j-1,2}$ as follows:

$$x_j = \left\lfloor \frac{\lfloor x_{j-1,1}2^\ell \rfloor}{2^\ell} \frac{\lfloor x_{j-1,2}2^\ell \rfloor}{2^\ell} 2^\ell \right\rfloor \frac{1}{2^\ell}$$
$$y_j = \left\lfloor \frac{\lfloor y_{j-1,1}2^\ell \rfloor}{2^\ell} \frac{\lfloor y_{j-1,2}2^\ell \rfloor}{2^\ell} 2^\ell \right\rfloor \frac{1}{2^\ell}.$$

We use ε_j for the error at level j on the product $x_j y_j$; in other words,

$$\left| \left\lfloor \frac{\lfloor x_j 2^\ell \rfloor}{2^\ell} \frac{\lfloor y_j 2^\ell \rfloor}{2^\ell} 2^\ell \right\rfloor \frac{1}{2^\ell} - x_j y_j \right| = \varepsilon_j$$

if and only if

$$\left| \left\lfloor \frac{\lfloor x_j 2^\ell \rfloor}{2^\ell} \frac{\lfloor y_j 2^\ell \rfloor}{2^\ell} 2^\ell \right\rfloor \frac{1}{2^\ell} - \frac{\lfloor x_{j-1} 2^\ell \rfloor}{2^\ell} \frac{\lfloor y_{j-1} 2^\ell \rfloor}{2^\ell} \right| = \varepsilon_j.$$

Before bounding ε_j from above, we notice that

$$\left| x_j - \frac{\lfloor x_j 2^\ell \rfloor}{2^\ell} \right| = \left| \left\lfloor \frac{\lfloor x_{j-1,1}2^\ell \rfloor}{2^\ell} \frac{\lfloor x_{j-1,2}2^\ell \rfloor}{2^\ell} 2^\ell \right\rfloor \frac{1}{2^\ell} - \frac{\lfloor x_{j-1,1}2^\ell \rfloor}{2^\ell} \frac{\lfloor x_{j-1,2}2^\ell \rfloor}{2^\ell} \right| = \varepsilon_{j-1}$$

and the same for y_j. We also have that

$$\begin{aligned} &\left| \frac{\lfloor x_j 2^\ell \rfloor}{2^\ell} \frac{\lfloor y_j 2^\ell \rfloor}{2^\ell} - x_j y_j \right| \\ &= \left| \frac{\lfloor x_j 2^\ell \rfloor}{2^\ell} \frac{\lfloor y_j 2^\ell \rfloor}{2^\ell} - x_j \frac{\lfloor y_j 2^\ell \rfloor}{2^\ell} + x_j \frac{\lfloor y_j 2^\ell \rfloor}{2^\ell} - x_j y_j \right| \\ &\le \left| \frac{\lfloor x_j 2^\ell \rfloor}{2^\ell} \frac{\lfloor y_j 2^\ell \rfloor}{2^\ell} - x_j \frac{\lfloor y_j 2^\ell \rfloor}{2^\ell} \right| + \left| x_j \frac{\lfloor y_j 2^\ell \rfloor}{2^\ell} - x_j y_j \right| \\ &= \left| \frac{\lfloor y_j 2^\ell \rfloor}{2^\ell} \right| \left| \frac{\lfloor x_j 2^\ell \rfloor}{2^\ell} - x_j \right| + |x_j| \left| \frac{\lfloor y_j 2^\ell \rfloor}{2^\ell} - y_j \right| \\ &= \varepsilon_{j-1} + \varepsilon_{j-1} = 2\varepsilon_{j-1}. \end{aligned}$$

Now we have that

$$\begin{aligned} \varepsilon_j &= \left| \left\lfloor \frac{\lfloor x_j 2^\ell \rfloor}{2^\ell} \frac{\lfloor y_j 2^\ell \rfloor}{2^\ell} 2^\ell \right\rfloor \frac{1}{2^\ell} - x_j y_j \right| \\ &= \left| \left\lfloor \frac{\lfloor x_j 2^\ell \rfloor}{2^\ell} \frac{\lfloor y_j 2^\ell \rfloor}{2^\ell} 2^\ell \right\rfloor \frac{1}{2^\ell} - \frac{\lfloor x_j 2^\ell \rfloor}{2^\ell} \frac{\lfloor y_j 2^\ell \rfloor}{2^\ell} + \frac{\lfloor x_j 2^\ell \rfloor}{2^\ell} \frac{\lfloor y_j 2^\ell \rfloor}{2^\ell} - x_j y_j \right| \\ &\le \left| \left\lfloor \frac{\lfloor x_j 2^\ell \rfloor}{2^\ell} \frac{\lfloor y_j 2^\ell \rfloor}{2^\ell} 2^\ell \right\rfloor \frac{1}{2^\ell} - \frac{\lfloor x_j 2^\ell \rfloor}{2^\ell} \frac{\lfloor y_j 2^\ell \rfloor}{2^\ell} \right| + \left| \frac{\lfloor x_j 2^\ell \rfloor}{2^\ell} \frac{\lfloor y_j 2^\ell \rfloor}{2^\ell} - x_j y_j \right| \\ &\le \frac{1}{2^\ell} + 2\varepsilon_{j-1}. \end{aligned}$$

If there are $m = \lceil \log_2 n \rceil$ levels,

$$\varepsilon_m \;\leq\; \frac{1}{2^\ell} \sum_{j=0}^{m} 2^j \;<\; \frac{2^{\lceil \log_2 n \rceil}}{2^{\ell-1}}.$$

It follows that a k-bit approximation of the product of n real numbers in interval $[0, 1]$, which corresponds to $\varepsilon_m \leq 2^{-k}$, is obtained if we truncate each intermediate subproduct to $\ell = k + 1 + \lceil \log_2 n \rceil$ bits.

ACKNOWLEDGEMENTS

The authors wish to thank Marc Kaplan and Nicolas Gisin for stimulating discussions about the simulation of entanglement. Furthermore, Marc has carefully read Ref. [17], in which the decomposition of the GHZ distribution as a convex combination of two sub-distributions was first accomplished, and he has pointed out that the lower bound from Ref. [22] applies even in the case of equatorial measurements. Alain Tapp pointed out that the entropy of the GHZ distribution can be as small as one bit in the case of measurements in the computational basis. The referees provided useful feedback.

REFERENCES

[1] A. Einstein, B. Podolsky, and N. Rosen, "Can quantum-mechanical description of physical reality be considered complete?" *Phys. Rev.*, vol. 47, pp. 777–780, May 1935.
[2] J. S. Bell, "On the Einstein Podolsky Rosen paradox," *Physics*, vol. 1, no. 3, pp. 195–200, 1964.
[3] T. Maudlin, "Bell's inequality, information transmission, and prism models," in *Proc. Biennial Meeting Philos. Sci. Assoc. (PSA)*, 1992, pp. 404–417.
[4] G. Brassard, R. Cleve, and A. Tapp, "Cost of exactly simulating quantum entanglement with classical communication," *Phys. Rev. Lett.*, vol. 83, pp. 1874–1877, Aug. 1999.
[5] M. Steiner, "Towards quantifying non-local information transfer: Finite-bit non-locality," *Phys. Lett. A*, vol. 270, no. 5, pp. 239–244, Jun. 2000.
[6] B. F. Toner and D. Bacon, "Communication cost of simulating Bell correlations," *Phys. Rev. Lett.*, vol. 91, Oct. 2003, Art. ID 187904.
[7] O. Regev and B. Toner, "Simulating quantum correlations with finite communication," *SIAM J. Comput.*, vol. 39, no. 4, pp. 1562–1580, 2009.
[8] N. J. Cerf, N. Gisin, and S. Massar, "Classical teleportation of a quantum bit," *Phys. Rev. Lett.*, vol. 84, no. 11, pp. 2521–2524, 2000.
[9] G. Brassard, "Quantum communication complexity," *Found. Phys.*, vol. 33, no. 11, pp. 1593–1616, 2003.
[10] D. M. Greenberger, M. A. Horne, and A. Zeilinger, "Going beyond Bell's theorem," in *Bell's Theorem, Quantum Theory and Conceptions of the Universe*, M. Kafatos, Ed. Dordrecht, The Netherlands: Kluwer, 1989, pp. 69–72.
[11] N. Gisin, private communication, 2010.
[12] J.-D. Bancal, C. Branciard, and N. Gisin, "Simulation of equatorial von Neumann measurements on GHZ states using nonlocal resources," *Adv. Math. Phys.*, vol. 2010, no. 2010, Dec. 2010, Art. ID 293245.
[13] C. Branciard and N. Gisin, "Quantifying the nonlocality of Greenberger–Horne–Zeilinger quantum correlations by a bounded communication simulation protocol," *Phys. Rev. Lett.*, vol. 107, Jul. 2011, Art. ID 020401.
[14] G. Brassard and M. Kaplan, "Simulating equatorial measurements on GHZ states with finite expected communication cost," in *Proc. 7th Conf. Theory Quantum Comput., Commun., Cryptogr. (TQC)*, 2012, pp. 65–73.
[15] D. E. Knuth and A. C. Yao, "The complexity of nonuniform random number generation," in *Algorithms and Complexity: New Directions and Recent Results*, J. F. Traub, Ed. New York, NY, USA: Academic, 1976, pp. 357–428.
[16] S. Massar, D. Bacon, N. J. Cerf, and R. Cleve, "Classical simulation of quantum entanglement without local hidden variables," *Phys. Rev. A*, vol. 63, no. 5, 2001, Art. ID 052305.
[17] C. Gravel, "Structure de la distribution de probabilité de l'état GHZ sous l'action de mesures de von Neumann locales," M.S. thesis, Dept. Comput. Sci. Oper. Res., Univ. Montreal, Montreal, QC, Canada, 2011. [Online]. Available: https://papyrus.bib.umontreal.ca/jspui/handle/1866/5511
[18] C. Gravel, "Structure of the probability distribution for the GHZ quantum state under local von Neumann measurements," *Quantum Phys. Lett.*, vol. 1, no. 3, pp. 87–96, 2012.
[19] J. von Neumann, "Various techniques used in connection with random digits," in *Monte Carlo Method*, Applied Mathematics Series, vol. 12. Gaithersburg, MD, USA: National Bureau of Standards, 1951, pp. 36–38.
[20] L. Devroye, *Non-Uniform Random Variate Generation*. New York, NY, USA: Springer, 1986.
[21] L. Devroye and C. Gravel. (2015). "Sampling with arbitrary precision." [Online]. Available: http://arxiv.org/abs/1502.02539
[22] A. Broadbent, P.-R. Chouha, and A. Tapp, "The GHZ state in secret sharing and entanglement simulation," in *Proc. 3rd Int. Conf. Quantum, Nano, Micro Technol.*, Feb. 2009, pp. 59–62.
[23] M. Kaplan, private communication, 2013.
[24] G. Brassard, L. Devroye, and C. Gravel, "Remote sampling with applications to general entanglement simulation," in preparation.
[25] W. Dür, G. Vidal, and J. I. Cirac, "Three qubits can be entangled in two inequivalent ways," *Phys. Rev. A*, vol. 62, no. 6, 2000, Art. ID 062314.
[26] C. Gravel, "Échantillonnage des distributions continues non uniformes en précision arbitraire et protocole pour l'échantillonnage exact distribué des distributions discrètes quantiques," Ph.D. dissertation, Dept. Comput. Sci. Oper. Res., Univ. Montreal, Montreal, QC, Canada, 2015. [Online]. Available: https://papyrus.bib.umontreal.ca/xmlui/handle/1866/12337

Gilles Brassard FRS, O.C., was born in Montreal, Quebec, Canada, in 1955. He obtained his Ph.D. from Cornell University in 1979, and joined the Département d'informatique et de recherche opérationnelle at Université de Montréal the same year, where he has been Canada Research Chair in quantum information science since 2001. His research interests include quantum information science, classical and quantum cryptography, and the foundations of quantum mechanics.

Luc Devroye was born in Tienen, Belgium. He obtained his Ph.D. from the University of Texas at Austin in 1976, and joined the School of Computer Science at McGill University in Montreal, Canada, the following year, where he is now James McGill Professor. His research interests include probability theory as applied to the analysis of algorithms, mathematical statistics, machine learning, pattern recognition, and random number generation.

Claude Gravel was born in Joliette, Quebec, Canada, in 1980. He obtained his Ph.D. from the Université de Montréal in 2015, where he is currently a postdoctoral fellow. He has been course lecturer at McGill University from 2012 until 2015, and a statistician at Hydro-Quebec from 2011 until 2012.