Mathematical Modeling of Quantum Repeaters Chains

Lorenzo La Corte

June 16, 2024

Mathematical Background

Convolution

Convolution is a fundamental mathematical operation used to combine two functions to produce a third function, which represents how the shape of one function is modified by the other.

For two discrete functions a and b, the convolution $a * b$ is defined as:

$$(a * b)(z) = \sum_{x=0}^{z} a(x) \cdot b(z - x) \tag{1}$$

Convolution of two Probability Distributions In the context of probability distributions, if X and Y are independent random variables with probability distribution functions p_X and p_Y, their sum $Z = X + Y$ has a probability distribution function p_Z given by the convolution of p_X and p_Y:

$$p_Z(z) = \sum_{x=0}^{z} p_X(x) \cdot p_Y(z - x) \tag{2}$$

Thus, convolution is used to determine the probability distribution of the sum of independent random variables by combining their individual probability distributions.

Example Consider this simple example. Let X and Y discrete independent random variables.

The probability distribution $\Pr(Z = z)$ of the random variable $Z = X + Y$ is computed as follows:

x	$\Pr(X=x)$	y	$\Pr(Y=y)$	z	$\Pr(Z=z)$	Derivation of $\Pr(Z=z)$
0	0.2	0	0.3	0	0.06	$0.2 \cdot 0.3$
1	0.5	1	0.4	1	0.23	$0.2 \cdot 0.4 + 0.5 \cdot 0.3$
2	0.3	2	0.3	2	0.35	$0.2 \cdot 0.3 + 0.5 \cdot 0.4 + 0.3 \cdot 0.3$
				3	0.27	$0.5 \cdot 0.3 + 0.3 \cdot 0.4$
				4	0.09	$0.3 \cdot 0.3$

Note that $\Pr(Z = z)$ has five elements (all the possible sums), and it is valid as its probabilities sum to 1.

The convolution operator $*$ is associative, meaning that for any three functions a, b, and c:

$$(a * b) * c = a * (b * c) \tag{3}$$

Random Variables

In this section, we fix notation on random variables and operations on them.

Most random variables in the context of quantum repeaters

- are discrete,
- have as domain a subset of nonnegative integers.

PDF Let X be such a random variable, then its probability distribution function is a map

$$p_X : x \mapsto \Pr(X = x) \tag{4}$$

which describes the probability that its outcome will be $x \in \{0, 1, 2, \ldots\}$.

CDF Equivalently, X is described by its cumulative distribution function

$$\Pr(X \leq x) = \sum_{y=0}^{x} \Pr(X = y), \tag{5}$$

which is transformed to the probability distribution function as

$$\Pr(X = x) = \Pr(X \leq x) - \Pr(X \leq x - 1). \tag{6}$$

Independent Random Variables Two random variables X and Y are independent if

$$\Pr(X = x \text{ and } Y = y) = \Pr(X = x) \cdot \Pr(Y = y) \tag{7}$$

for all x and y in the domain.

Copies of a Random Variable By a *copy* of X, we mean a fresh random variable which is independent from X and identically distributed (i.i.d.). We will denote a copy by a superscript in parentheses. For example, $X^{(1)}$, $X^{(142)}$ and $X^{(A)}$ are all copies of X.

The mean of X is denoted by

$$E[X] = \sum_{x=0}^{\infty} \Pr(X = x) \cdot x \tag{8}$$

and can equivalently be computed as

$$E[X] = \sum_{x=1}^{\infty} \Pr(X \geq x). \tag{9}$$

Function of Random Variables If f is a function which takes two nonnegative integers as input, then the random variable $f(X, Y)$ has probability distribution function defined as

$$\Pr(f(X, Y) = z) := \sum_{\substack{x=0, y=0 \\ f(x,y)=z}}^{\infty} \Pr(X = x \text{ and } Y = y). \tag{10}$$

Sum of Random Variables An example of such a function is addition.

Define $Z := X + Y$ where X and Y are independent, then the probability distribution p_Z of Z is given by

$$p_Z(z) = \Pr(Z = z) = \sum_{\substack{x=0, y=0 \\ x+y=z}}^{\infty} \Pr(X = x \text{ and } Y = y). \tag{11}$$

But since $y = z - x$ this is equivalent to

$$p_Z(z) = \Pr(Z = z) = \sum_{x=0}^{z} \Pr(X = x \text{ and } Y = z - x) \tag{12}$$

$$= \sum_{x=0}^{z} \Pr(X = x) \cdot Pr(Y = z - x) \tag{13}$$

$$= \sum_{x=0}^{z} p_X(x) \cdot p_Y(z - x) \tag{14}$$

which is the convolution of the distributions p_X and p_Y, denoted as $p_Z = p_X * p_Y$ (see convolution).

Since convolution operator $*$ is associative, writing $a * b * c$ is well-defined, for functions a, b, c from the nonnegative integers to the real numbers (see associativity of convolution).

In general, **the probability distribution of sums of independent random variables equals the convolutions of their individual probability distribution functions**.

Geometric Distribution

The Geometric Distribution is a discrete probability distribution that models the number of trials needed to achieve the first success in a sequence of independent Bernoulli trials, each with the same success probability p.

Probability Distribution Function (PDF)

The Probability Distribution Function (PDF) of a Geometric Distribution gives the probability that the first success occurs on the t-th trial. It is defined as:

$$\Pr(T = t) = p(1 - p)^{t-1} \quad \text{for} \quad t \in \{1, 2, 3, \ldots\}, \tag{15}$$

where:

- T is the random variable representing the trial number of the first success,
- p is the probability of success on each trial,
- $(1 - p)$ is the probability of failure on each trial.

This formula expresses that the first $t - 1$ trials must be failures (each occurring with probability $1 - p$), and the t-th trial must be a success (with probability p).

Cumulative Distribution Function (CDF)

The Cumulative Distribution Function (CDF) of a Geometric Distribution gives the probability that the first success occurs on or before the t-th trial. It is defined as:

$$\Pr(T \leq t) = 1 - (1-p)^t. \tag{16}$$

Derivation of the CDF This is the derivation of the CDF of a Geometric Distribution, from its PDF

$$\begin{aligned}
\Pr(T \leq t) &= 1 - \Pr(T > t) \\
&= 1 - \sum_{k=t+1} \Pr(T = k) \\
&= 1 - \left\{ p(1-p)^t + p(1-p)^{t+1} + p(1-p)^{t+2} + \dots \right\} \\
&= 1 - p(1-p)^t \sum_{k=0} (1-p)^k \\
&= 1 - (1-p)^t \sum_{k=0} p(1-p)^k \\
&= 1 - (1-p)^t.
\end{aligned}$$

This CDF formula indicates the probability that the first success occurs within the first t trials.

Mathematical Model for Waiting Time and Fidelity

We derive expressions for the waiting time and fidelity of the first generated end-to-end link in the repeater chain protocol.

We derive a recursive definition for the random variable T_n, which **represents the waiting time in a $2n$-segment repeater chain**.

Extending this definition to the Werner parameter W_n of the pair, which stands in one-to-one correspondence to its fidelity F_n using the equation:

$$F_n = \frac{1 + 3W_n}{4}. \tag{17}$$

Note that all operations in the repeater chain protocols we study

- Entanglement generation over a single hop
- Distillation
- Swapping

take a duration that is a multiple of L_0/c, the time to send information over a single segment.

For this reason, it is common to denote the waiting time in **discrete units** of L_0/c, which is a convention we comply with for T_n. Cutoffs as well follow the same discrete units.

Heraldeld Entanglement Generation

Waiting Time for Elementary Entanglement

In modeling the random variable T_n, which represents the waiting time in a 2^n segment repeater chain, we can reason by induction.

The base case T_0 **is the waiting time for the generation of elementary entanglement**.

Since we model the generation of single-hop entanglement by attempts which succeed with a fixed probability p_{gen}, the waiting time T_0 is a discrete random variable (in units of L_0/c) which follows a geometric distribution with probability distribution given by

$$\Pr(T_0 = t) = p_{\text{gen}}(1 - p_{\text{gen}})^{t-1} \quad \text{for} \quad t \in \{1, 2, 3, \ldots\}. \tag{18}$$

For what follows, it will be more convenient to specify T_0 by its cumulative distribution function (CDF), which is given by

$$\Pr(T_0 \leq t) = 1 - (1 - p_{\text{gen}})^t. \tag{19}$$

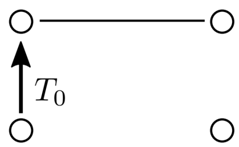

Figure 1: For two segments, T_0 represents the waiting time for the generation of a single link between two nodes without any intermediate repeater nodes.

Werner Parameter for Elementary Entanglement

The output state for the generation is a Werner state with Werner parameter w_0.

Numerical Examples

To plot the CDF and PDF of the waiting time for the entanglement generation protocol, we can use the following code snippet:

```python
def entanglement_generation(p_gen=0.5, t_trunc=20):
    parameters = {
        # A protocol is represented by a tuple of 0 and 1,
        # where 0 stands for swap and 1 stands for distillation.
        # This example is a 0-level swap protocol,
        # as we consider only the entanglement generation.
        "protocol": (),
        # success probability of entanglement generation
        "p_gen": p_gen,
        # truncation time for the repeater scheme.
        # It should be increased to cover more time step
        # if the success proability decreases.
        # Commercial hardware can easily handle up to t_trunc=1e5
        "t_trunc": t_trunc,
        "w0": 0.98, # ignore this for the sake of the example
        "p_swap": 0.5  # ignore this as well
    }
    pmf, _ = repeater_sim(parameters)
    return pmf
```

The following figures show the CDF and PDF of the waiting time for the entanglement generation protocol, with $p_{\text{gen}} = 0.2$ and $p_{\text{gen}} = 0.5$. We can see that the waiting time is **distributed according to a geometric distribution**, as expected.

In order for the algorithm to be computationally feasible, **the waiting time is truncated**, in this case at $t = 20$. The parameter t_{trunc} in the algorithm should chosen to be large enough to capture the bulk of the distribution, but small enough to keep the computation time reasonable. The truncation brings to an approximation of the distribution, as the probability of waiting for more than 20 units of time is not considered: this is more visible in the top plot, where the CDF is less close to 1 at $t = 20$.

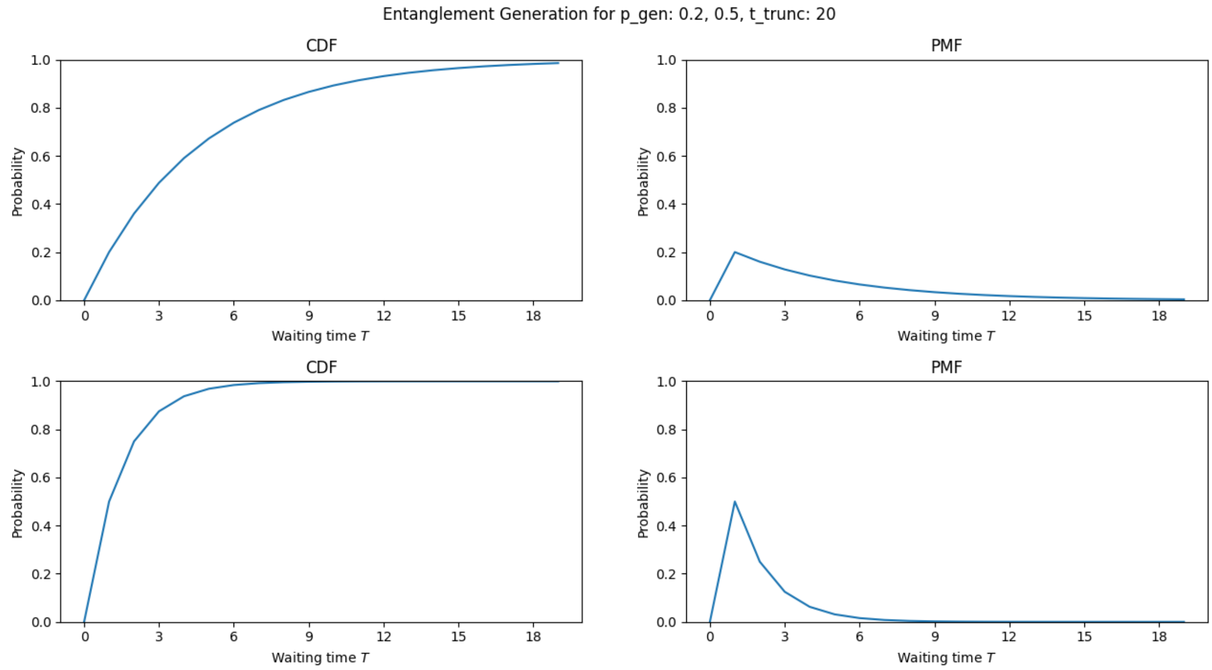

Figure 2: CDF and PDF of the waiting time for the entanglement generation protocol, with $p_{gen} = 0.2$ (top) and $p_{gen} = 0.5$ (bottom).

Entanglement Swapping

Once we have generated elementary entanglement, we can use it to **create entanglement over longer distances by entanglement swapping**. We defined T_0 as the waiting time for the generation of elementary entanglement, and our base for the induction. We now define our inductive step assuming that we have found an expression for T_n and we want to construct T_{n+1}.

In order to perform the entanglement swap to produce a single $(2^n + 1)$-hop link, a node needs to wait for the production of two (2^n)-hop links, one on each side.

Denote the waiting time for the pairs by $T_n^{(A)}$ and $T_n^{(B)}$, both of which are i.i.d. with T_n, as they are copies of it.

Waiting Time for Entanglement Swapping

Time until both pairs are available

We introduce a new random variable M_n modeling the time until both pairs are available

$$M_n := \max(T_n^{(A)}, T_n^{(B)}) \tag{20}$$

which is also **the time at which the entanglement swap ends**.

This is distributed according to the CDF

$$\Pr(M_n \leq t) = \Pr(T_n^{(A)} \leq t \text{ and } T_n^{(B)} \leq t) = \Pr(T_n \leq t)^2. \tag{21}$$

From it, we can derive the PDF

$$\Pr(M_n = t) = \sum_{\substack{t_A, t_B: \\ t = \max(t_A, t_B)}} \Pr(T_n^{(A)} = t_A \text{ and } T_n^{(B)} = t_B). \tag{22}$$

Number of steps required

We introduce now K_n, a random variable following a geometric distribution

$$\Pr(K_n = k) = p_{swap}(1 - p_{swap})^{k-1} \tag{23}$$

modeling **the number of steps k until the first successful swap** at level n.

The fact that it follows a geometric distribution is a direct consequence of our choice to model the success probability p_{swap} to be independent of the state of the two input links.

Derivation of T_{n+1}

The derivation of T_{n+1} requires us to combine the random variable for the number of steps required K_n and the random variable for the waiting time for one attempt M_n.

In order to find the relation between M_n and T_{n+1}, note that when entanglement swap fails, the two input are lost and need to be regenerated. The regeneration of fresh entanglement after each failing entanglement swap **adds to the waiting time**.

Thus, T_{n+1} is a *compound random variable*: it is the sum of K_n copies of M_n.

Since the number of entanglement swaps K_n is geometrically distributed, we say that T_{n+1} is a *geometric compound sum* of K_n copies of M_n, denoted as

$$T_{n+1} = \sum_{k=1}^{K_n} M_n^{(k)} \tag{24}$$

Derivation of the PDF of T_{n+1}

The probability distribution of the waiting time T_{n+1}

$$\Pr(T_{n+1} = t) = \Pr\left[\sum_{k=1}^{K_n} M_n^{(k)} = t\right]$$

is computed as the marginal of the waiting time conditioned on a fixed number of swaps

$$\Pr(T_{n+1} = t) = \sum_{k=1}^{\infty} \Pr(K_n = k) \cdot \Pr\left[\left(\sum_{j=1}^{k} M_n^{(j)}\right) = t\right]. \tag{25}$$

Plugging in the expressions for $\Pr(K_n = k)$ (...) and recalling that the sum of k copies of M_n can be expressed as a convolution, we get

$$\Pr(T_{n+1} = t) = \sum_{k=1}^{\infty} p_{\text{swap}}(1 - p_{\text{swap}})^{k-1} \left(*_{j=1}^{k} m\right) \tag{26}$$

where, from (...)

$$m(t) := \Pr(M = t) = \sum_{\substack{t_A, t_B: \\ t = \max(t_A, t_B)}} \Pr(T_n^{(A)} = t_A \text{ and } T_n^{(B)} = t_B).$$

Werner Parameter for Entanglement Swapping

Considering two entangled pairs, respectively with werner parameters w_A and w_B, the output werner parameter w_{out}, if we do not consider decoherence, will be

$$w_{out} = w_A \cdot w_B. \tag{27}$$

However, when the first of the two pairs is generated, it has to wait for the elementary generation of the other; **during this time the first generated pair decoheres**. In particular, a Werner state $\rho(w)$ residing in memory for a time $\Delta(t)$ will transform into the Werner state $\rho(w_{decayed})$ with

$$w_{\text{decayed}} = w \cdot e^{-\Delta t / T_{coh}} \tag{28}$$

where T_{coh} is the joint coherence time of the two quantum memories holding the qubits.

Denote by A and B the input links to the entanglement swap and denote by (t_A, w_A) and (t_B, w_B) their respective delivery times and Werner parameters. Without loss of generality, suppose that the link A is produced after link B, i.e. $t_A \geq t_B$.

Link A is produced last, so the entanglement swap will be performed directly after its generation and hence link A will enter the entanglement swap with Werner parameter w_A. Link B is produced earliest and will therefore decohere until production of link A.

It follows that B's Werner parameter decoheres accordingly to (??): w_B, immediately before the swap, is equal to

$$w'_B = w_B \cdot e^{-|t_A - t_B|/T_{coh}}.$$

Thus, the entanglement swap would produce the 2^{n+1}-hop state with Werner parameter

$$\begin{aligned} w_{out} &= w_A \cdot w'_B \\ &= w_A \cdot w_B \cdot e^{-|t_A - t_B|/T_{coh}}. \end{aligned} \tag{29}$$

Notice that choosing $t_A > t_B$ would have lead to the same result.

In order to model at the same time the Werner Parameter and the Waiting Time we use a joint random variable (T_n, W_n).

For a single segment ($n = 0$), we are in the, already discussed, entanglement generation phase. Here, the waiting time and Werner parameter are uncorrelated because we model the attempts at generating single-hop entanglement to be independent and to each take equally long.

At the recursive step, we model the waiting time and Werner parameter as a joint random variable (T_n, W_n) as

$$(T_{n+1}, W_{n+1}) := \sum_{k=1}^{K_n} (M_n, V_n)^{(k)}. \tag{30}$$

The auxiliary joint random variable (M_n, V_n) is defined as

$$(M_n, V_n) := g\left((T_n, W_n)^{(A)}, (T_n, W_n)^{(B)}\right). \tag{31}$$

The function g is given by

$$g\left((t_A, w_A), (t_B, w_B)\right) := \left(g_T(t_A, t_B), g_W\left((t_A, w_A), (t_B, w_B)\right)\right), \tag{32}$$

where g_T is defined in eq. 10 and

$$g_W\left((t_A, w_A), (t_B, w_B)\right) := w_A \cdot w_B \cdot e^{-\frac{|t_A - t_B|}{T_{\text{coh}}}} \tag{33}$$

with T_{coh} the quantum memory coherence time.

We already studied the expression for the waiting time T_{n+1}. The other random variable W_{n+1} directly derives from V_n, which expresses the Werner parameter of the produced 2^{n+1}-hop link in case the swap is successful.

To prove that V_n is formulated properly, we argue that g_W correctly computes the Werner parameter of the output link after an entanglement swap.

If the entanglement swap fails, then the 2^{n+1}-hop link with its Werner parameter will never be produced since both initial 2^n-hop entangled pairs are lost. Instead, two fresh 2^n-hop links will be generated.

In order to find how the Werner parameter on level $n+1$ is expressed as a function of the waiting times and Werner parameters at level n, consider a sequence (m_j, v_j) of waiting times m_j and Werner parameters v_j, where j runs from 1 to the first successful swap k.

- m_j correspond to the waiting time until the end of the entanglement swap that transforms two 2^n-hop links into a single 2^{n+1}-hop link
- v_j to the output link's Werner parameter if the swap were successful.

From previus results, we found that the total waiting time is given by $\sum_{j=1}^{k} m_j$, the sum of the duration of the production of the lost pairs (see ??).

Note, however, that the Werner parameter of the 2^{n+1} hop link **is only influenced by the links that the successful entanglement swap acted upon**. Since the entanglement swaps are performed until the first successful one, the output link is the last produced link and therefore its Werner parameter equals v_k.

We thus find that the waiting time t_{final} of the first 2^{n+1}-hop link and its Werner parameter w_{final} are given by:

$$t_{final} = \sum_{j=1}^{k} m_j \tag{34}$$

$$w_{final} = v_k \tag{35}$$

or, in a more compact form

$$(t_{final}, w_{final}) = \left(\sum_{j=1}^{k} m_j, v_k \right). \tag{36}$$

Taking into account that the number of swaps k that need to be performed until the first successful one is an instance of the random variable K_n, we arrive at the full recursive expression for the waiting time and Werner parameter at level $n+1$ as given in eq. (??).

Numerical Examples