Runtime–coherence trade-offs for hybrid SAT-solvers

Vahideh Eshaghian(1), Sören Wilkening(2),(3), Johan Åberg(1), David Gross(1)*
(1) Institute for Theoretical Physics, University of Cologne,
Germany; (2) Institut für Theoretische Physik, Leibniz Universität Hannover,
Germany; (3) Volkswagen AG, Berliner Ring 2, 38440 Wolfsburg;

(Dated: April 24, 2024)

Many search-based quantum algorithms that achieve a theoretical speedup are not practically relevant since they require extraordinarily long coherence times, or lack the parallelizability of their classical counterparts. This raises the question of how to divide computational tasks into a collection of parallelizable sub-problems, each of which can be solved by a quantum computer with limited coherence time. Here, we approach this question via hybrid algorithms for the k-SAT problem. Our analysis is based on Schöning's algorithm, which solves instances of k-SAT by performing random walks in the space of potential assignments. The search space of the walk allows for "natural" partitions, where we subject only one part of the partition to a Grover search, while the rest is sampled classically, thus resulting in a hybrid scheme. In this setting, we argue that there exists a simple trade-off relation between the total runtime and the coherence-time, which no such partition based hybrid-scheme can surpass. For several concrete choices of partitions, we explicitly determine the specific runtime coherence-time relations, and show saturation of the ideal trade-off. Finally, we present numerical simulations which suggest additional flexibility in implementing hybrid algorithms with optimal trade-off.

I. INTRODUCTION

Consider a quantum algorithm that takes exponential time to run, but still offers a polynomial speedup over the best classical method. Examples include Grover searches to brute-force a password or for finding the solution for a combinatorial optimization problem for which no classical heuristics exist. Fully quantum implementations might not be desirable for two reasons: (1) Quantum hardware that can sustain very long computations might not be available, and (2) quantum algorithms, like Grover's search, might not be easily amenable to parallelization. One is thus lead to the question of how to best break up such instances into a set of smaller, parallelizable subproblems that can individually be solved on quantum hardware.

We consider the well-known *satisfiability problem* with k the number of literals in each clause, (k-SAT) and focus particularly on 3-SAT since it provides an attractive test bed to investigate such questions. k-SAT is the archetypical combinatorial optimization problem and represents a class of use cases with considerable practical relevance. Moreover, there is a classical randomized algorithm [1, 2] due to Schöning, with a performance close to the best-known algorithms with provable performance, and which furthermore allows for a closed-form asymptotic run-time analysis. And indeed, the algorithm obtained by replacing the classical search of the Schöning-procedure by a Grover search [3] yields a *quantum-Schöning algorithm* with a quadratic improvement *vis-à-vis* its classical counterpart [4]. (Below, we will refer to quantum algorithms that arise this way as *Groverizations* of their classical versions).

However, such 'fully quantized' Schöning's SAT-solvers cannot be performed in parallel, which arguably is a relevant feature for algorithms that run in exponential time. Hybrid schemes, based on 'partial' Groverizations of Schöning's algorithm, where Grover search procedures are applied only to certain sub-routines, usually do allow for parallelizations.

Starting point of our analysis is the stochastic nature of Schöning's algorithm as a random walk. This point of view yields two classes of hybrid algorithms, where one class Groverizes the random choice of the initial state of the walk, while the other class Groverizes the randomness in the walk itself. Within an established model of Schöning's algorithm, we optimize the resulting run-times by balancing the resources allocated to the subroutines.

A. Runtime–Coherence Time Trade-Offs

Before specializing the Schöning process, let us briefly outline the trade-offs between runtime and coherence time that can be expected for quantum search problems. Consider an algorithm that solves instances of size n with runtime $T(n)$. For exponential-time algorithms, we work with a somewhat coarser measure, the *(asymptotic) runtime rate*

$$\gamma = \lim_{n\to\infty} \frac{1}{n} \log T(n),$$

*Electronic address: vahideh@thp.uni-koeln.de where we drop the base of the logarithm from here on; the base is 2 unless explicitly stated otherwise. In other words, $T \in O^*(2^{\gamma n})$, where O^* denotes scaling behavior up to polynomial factors. The aim is to trade it off against the *coherence time* required to run the algorithm. If $C(n)$ is the longest time over which coherence has to be maintained while running the algorithm, then the *coherence time rate* is

$$\chi = \lim_{n\to\infty} \frac{1}{n} \log C(n).$$

Now restrict attention to search algorithms with classical runtime rate γ_{C}. A completely Groverized version runs with rate $\gamma_{\mathrm{G}} = \gamma_{\mathrm{C}}/2$. All of its runtime will be spent coherently, specifically executing Grover iterations. Therefore, $\chi_{\mathrm{G}} = \gamma_{\mathrm{C}}/2$ as well. We can visualize these two points in a "runtime rate vs coherence time rate"-chart, a mode of visualization that we will employ frequently (Fig. 1).

To achieve a trade-off between total runtime and coherence time, we will consider algorithms that apply Grover's procedure only to a subset of the search space. It is easy to see that any algorithm which results from such a procedure must have coordinates (χ, γ) that lie on or above the line segment

$$L = \{(\chi, \gamma_{\mathrm{C}} - \chi) \,|\, \chi \in [0, \gamma_{\mathrm{C}}/2]\}$$

that connects the purely classical point $(0, \gamma_{\mathrm{C}})$ to the completely Groverized one $(\gamma_{\mathrm{C}}/2, \gamma_{\mathrm{C}}/2)$.

Indeed, take a partial Groverization that achieves parameters (χ, γ). Then one can replace the Grover part by a classical search. The resulting classical algorithm will have parameters $(0, \gamma + \chi)$, because the Grover search contributed χ to the runtime rate, but its classical simulation will contribute 2χ instead. But if the initial parameters were below the line, i.e. if $\gamma < \gamma_{\mathrm{C}} - \chi$, then the resulting classical algorithm runtime rate is $\gamma + \chi < \gamma_{\mathrm{C}}$, contradicting the assumption that γ_{C} describes the classical complexity of the search.

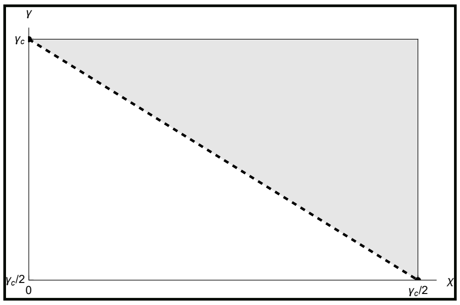

FIG. 1: We will frequently visualize the behavior of algorithms by indicating their position in a "runtime rate vs coherence time rate"-chart. Classical algorithms require no coherence, and thus lie on the y-axis. In the example given, the point on the upper left hand side represents a classical probabilistic search with runtime rate γ_{C}. A completely Groverized version has coordinates $(\gamma_{\mathrm{C}}/2, \gamma_{\mathrm{C}}/2)$ (bottom right), meaning that it will spend its entire runtime coherently. Hybrid algorithms that use Grover only for a subset of the search space must lie in the shaded area above or on the dashed line segment connecting these two points.

It is not obvious that, conversely, every point on this optimal line segment can actually be realized, much less with an algorithm that is "natural" or easy to implement. Deciding the parameter ranges for natural partial Groverizations of Schöning's procedure is the main goal of this paper.

1. *Related work*

Dunjko et al. [5] have previously considered partial Groverizations of Schöning's algorithm. They aimed to minimize a different metric: total number of clean qubits, rather than coherence time. In fact, they work in a highly constrained regime, where the number of available clean qubits only scales as cn, with $0 < c < 1$ and n the number of variables of the given 3-SAT formula. Surprisingly, they show that even this meager allotment of qubits in principle yields a speed-up compared to the classical Schöning's algorithm [12].

Despite the superficial similarities, their and our papers are quite different. We allow for qubit-counts that are quasi-linear in n, i.e. $O(n \log n)$, reasoning that for exponential-time algorithms, coherence time and parallelizability might be more limiting than the number of available qubits. As it will turn out, the setting considered here can interpolate between the classical and the fully Groverized performance, while the runtime rates obtainable in [5] stay close to the classical ones. While [5] uses de-randomization techniques, our approach builds more directly on the original Schöning's algorithm. This makes our approach technically less involved, and it also makes the lessons learned more widely applicable, since the basic technique of using Grover search over a subset of all variables, directly generalizes to any NP problem, whereas de-randomizations to a larger extent rely on the particular structure of the problem at hand.

II. SETTING THE STAGE

A. Schöning's algorithm

Here we provide a very brief introduction to the pertinent aspects of Schöning's 3-SAT solver. For a more thorough review, we refer the reader to [1, 2]. In the 3-SAT problem, we are given a collection of *clauses* $C_1, \ldots, C_L$ on n binary variables, where each clause is of the form $C_j = l_0^{(j)} \vee l_1^{(j)} \vee l_2^{(j)}$, and where each of the *literals* $l_0^{(j)}, l_1^{(j)}, l_2^{(j)}$ is one of the binary variables or its negation. The 3-SAT formula is the conjunction of all the given clauses, $C := \wedge_{j=1}^{L} C_j$, and the computational task is to determine whether there exists an assignment of the n binary variables that satisfies C. According to Schöning [1], an algorithm exists that, although with run-time that is exponential in n, can perform better than an exhaustive search through all potential assignments.

Schöning's algorithm (Alg. 1) depends on two parameters N, m to be determined later. It begins by choosing an assignment $x \in \{0,1\}^{\times n}$ uniformly at random. The algorithm then performs an m-step *random walk* over the space of n-bit strings (the inner loop in Alg. 1, from Line 5). In every step, it checks (according to a pre-determined order) all the clauses $C_1, \ldots, C_L$. If all are satisfied, then x is a solution and the algorithm terminates. Otherwise, it finds the first unsatisfied clause, chooses one of the three variables corresponding to the literals of that clause uniformly at random. The value of x is then updated, by negating that variable. This concludes the step. If no solution is found after m steps, the walk is terminated. Up to N such walks are attempted (the outer loop in Alg. 1), each time using a fresh uniformly random starting point x.

Algorithm 1 Schöning's Algorithm

```
1:  function SCHOENING($C_1, \ldots, C_L, N, m$)
2:      for $i = 1 \ldots N$ do
3:          $x \leftarrow$ uniformly random value from $\{0,1\}^{\times n}$
4:          for $j = 1 \ldots m$ do
5:              if $x$ satisfies $C_1, \ldots, C_L$ then
6:                  return $x$
7:              else
8:                  $k \leftarrow$ index of first unsatisfied clause
9:                  $l \leftarrow$ index of one of the three variables occurring in $C_k$, chosen uniformly at random
10:                 $x \leftarrow x$, with the $l$-th bit of $x$ flipped
11:             end if
12:         end for
13:     end for
14:     return False
15: end function
```

B. Analysis of the run-time of Schöning's algorithm

The analysis of the run-time of Schöning's algorithm is sketched in [1, 2] and a more in-depth analysis can be found in [6]. Here we follow a very similar line of reasoning, with our particular ansatz in mind. In the following, we present an overview, see Appendix A for a more detailed account.

Assume that there is at least one satisfying assignment $x^\star$. We first aim to lower-bound the probability that a given random walk finds a solution. Let x_0 be the (random) initial configuration, and x_l the one attained after the l-th step of the random walk. The probability that *any* solution is found during *any* step of the walk is certainly at least as large as the probability $P(x_m = x^\star)$ that the walk finds $x^\star$ at the m-th step. To analyze $P(x_m = x^\star)$, we follow in the steps of Schöning [1, 2], and focus on the evolution of the *Hamming-distance* $d_H(x_l, x^\star)$ between the current configuration and the selected satisfying assignment $x^\star$.

The fundamental insight is that if a clause C_k is violated at the l-th step, then at least one of the three variables that appear in C_k must differ between x_l and the satisfying assignment $x^\star$. Thus, the random flip decreases the Hamming distance to the solution with probability at least $1/3$:

$$P\big(d_H(x_{l+1}, x^\star) = d_H(x_l, x^\star) - 1\big) \geq \frac{1}{3}. \tag{1}$$

This suggests to pass from a description of the process on bit strings to its projection $x_l \mapsto d_H(x_l, x^\star)$ onto $\mathbb{N}$. However, this would generally yield a process that would be no easier to analyze than the original one. One may for example note that although Schöning-process $(x_l)_l$ is Markovian on the space of bit-strings $\{0,1\}^{\times n}$, one cannot generally expect its projection $(d_H(x_l, x^\star))_l$ to be Markovian on $\mathbb{N}$.

The general idea for the analysis is to replace (via a coupling) the true projection $(d_H(x_l, x^\star))_l$ with another process $(d_l)_l$ on $\mathbb{Z}$, which is Markovian and which moreover upper-bounds the true Hamming-distance,

$$d_H(x_l, x^\star) \leq d_l. \tag{2}$$

More precisely, the Markov process $(d_l)_l$ is defined by the transition probabilities

$$P(d_{l+1} = d_l + 1) = \frac{2}{3}, \quad P(d_{l+1} = d_l - 1) = \frac{1}{3}. \tag{3}$$

The transition probabilities (3) can be interpreted as worst-case scenarios of each step in the Schöning process. From the bound (2) it follows that $P(x_l = x^\star) \geq P(d_l \leq 0)$. In other words, the success probability of the Schöning-process is lower-bounded by the probability that the substitute-process d_l reaches 0.

Given the lower bound $P(d_m \leq 0)$ on the probability of success of each given walk, we expect at least one out of $N = 1/P(d_m \leq 0)$ walks to find $x^\star$. More precisely, if ϵ is the tolerated probability for failure, then the number of repetitions needed in order to find an existing solution satisfies

$$N \geq \frac{\log \epsilon}{\log(1 - P(d_m \leq 0))}. \tag{4}$$

The required number N of repetitions will be exponential in n. It is then common to take a coarser point of view, and only analyze the corresponding *rate*

$$\gamma := -\lim_{n\to\infty} \frac{1}{n} \log P(d_m \leq 0), \qquad \text{so that} \qquad N = O^*(2^{\gamma n}), \tag{5}$$

where O^* denotes scaling behavior up to polynomial factors in order to achieve any constant probability of failure ϵ. With the choice $m = n$ (i.e., the termination time is equal to the number of variables) it turns out [1, 2] that $\gamma < \log \frac{4}{3} \approx 0.415$.

It is surprisingly technically difficult to rigorously derive the "global bound" $P(x_l = x^\star) \geq P(d_l \leq 0)$ from the "local bound" (1). However, the Markovian version $(d_l)_l$ of the Hamming distance random walk is commonly accepted as a good (in fact, conservative) model of the Schöning-process. In the main body of this paper, we will therefore phrase our arguments in terms of that model. More technical details on the relation between the two processes are given in Appendix A.

C. Partial Groverizations: The general idea

For random walks we naturally tend to think of the randomness as being generated whenever needed, like when we assign the initial state, or make the random choices along the path. However, we can alternatively picture the walk as a deterministic process that is fed with an external random string S; a list from which it picks the next entry whenever a random choice is to be made. When the purpose of the walk is to find (an efficiently recognizable) solution to some computational problem, one can thus view the walk as a (deterministic) map that designates each input string S as being "successful" or "unsuccessful", in the sense of the walk reaching the satisfying solution $x^\star$ or not. To this mapping, we can in principle apply a Grover-search procedure, since the walk (as well as the solution-recognition procedure) can be performed via reversible circuitry, and can thus also be implemented coherently.

As described in the previous section, Schöning's algorithm proceeds with an initialization, followed by a random walk on the space of 2^n assignments. The initialization requires n bits of randomness, S_I, since the initial state is selected uniformly over all 2^n strings. A walk of length m requires a string S_W of $m \log 3$ bits to encode the needed randomness. The $\log 3$-factor is due to the fact that, at each step, the algorithm randomly selects which one of the three literals (of the first violated clause) should be flipped. An m-step Schöning-walk can thus be viewed as a map from $S = (S_I, S_W)$ to a binary variable that tells us whether a satisfying assignment has been reached or not.

With a coherent circuit that implements this map, we can thus replace the uniformly distributed random variable S, with a uniform superposition over a corresponding number of qubits, and proceed via standard Grover-iterations [3]. We would expect such a procedure to yield a satisfying assignment at a run-time that scales as $O^*(2^{n\gamma_G})$ iterations, with $\gamma_G = \frac{1}{2} \log \frac{4}{3} \approx 0.208$ [4], i.e., the standard quadratic speed-up. Up to a few constant qubits, one needs $n + (\log 3 + \log L)m$ qubits to encode this map as a quantum circuit, where L is the number of clauses in the 3-SAT formula (more details are given in Section V). Since the number of clauses grows linearly in n for the regime of interest by the SAT phase-transition conjecture [7], and for the Schöning walk $m = n$, the space complexity of such encoding is $O(n \log n)$.

The view of random walks as maps on random input strings opens up for the concept of partial Groverizations. Nothing would in principle prevent us from regarding only a *part* of the input string S as the input of the Grover-procedure, while keeping the rest of the string classical. Needless to say, one would generally expect the result to be less efficient than the "full" Groverization. However, the gain would be that the partial Groverization breaks the tasks into a collection of subproblems, each of which can be run in parallel on a quantum device that requires shorter coherence time.

Although it seems reasonable to expect that such a division in principle is always possible, one may also expect that it in general would be challenging to find a quantum circuit that implements it in an economical manner. (We can always resort to a full coherent circuit for S in its entirety, putting the "classical part" in a diagonal state.) However, there may be "natural" divisions of the process, which can be exploited. For Schöning's algorithm it is close to hand to consider the division $S = (S_I, S_W)$, i.e., the division of the required randomness into the initialization-part and the walk-part. One can thus consider two particularly natural classes of "partial" Groverizations of Schöning's algorithm. For one of these, the *Groverized Initialization (GI)*, the choice of the initial state is implemented coherently, while the walk is kept "classical". For the *Groverized Walk (GW)*, the choice of initial state is kept classical, while the walk itself is performed coherently.

As described in Section II B, the actual analysis is based on the random walk $(d_l)_l$ on $\mathbb{Z}$, rather than the true Schöning walk on strings in $\{0,1\}^{\times n}$. The idea is nevertheless the same; the required randomness is divided into the initialization and the walk *per se*, resulting in GI- and GW-processes. As described in Section II B, the rate of the true Schöning-process can be bounded by the rate of the substitute process $(d_l)_l$. It turns out that a similar argument can be made for GW (see Appendix A), thus yielding a rigorous bound for the rate also in this case. However, for the other processes we rather regard the $(d_l)_l$ process as a model of the genuine Schöning-walk, without rigorous guarantees of analogous bounds.

III. PARTIAL GROVERIZATIONS

The previous section introduced two types of partial Groverizations of Schöning's algorithm, GI and GW, based on the division $S = (S_I, S_W)$, i.e. the initial and the walk randomness. In this section, we describe these schemes in detail and further discuss their "fractional" cases.

In the GI scheme, there is an outer loop that classically samples S_W, and is followed by a Grover-search inner loop over the space of all possible S_I. Similarly, GW starts with a classical outer loop that samples S_I and is followed by a Grover-search inner loop over the space of all possible S_W (this space is well-defined as the walk length is fixed). We obtain Fractional Groverized Initialization (FGI) by adapting GI to a regime where only a fraction z of the variables in the initialization can be searched coherently, with $0 \leq z \leq 1$. Fractional Groverized Walk (FGW) is similarly an adaption of GW to a regime where Grover-search can be performed on the randomness of walks of at most m_q steps, with $0 \leq m_q$. In both these fractional schemes, two classical outer loops contain a Grover-search inner loop. The algorithms introduced here depend on parameters (N_1, N_2, etc), that will be specified explicitly in Section IV.

All Grover searches will use an oracle derived from the function shown in Alg. 2: It tests whether a Schöning-walk with initial configuration $x \in \{0,1\}^n$ and walk randomness $w \in \{1,2,3\}^m$ will lead to a satisfying assignment. For notational convenience, we let the elements of w take ternary in values, with the interpretation that w_l determines which of the three literals occurring in the first violated clause (if any) in step l of the walk is flipped. For a qubit-based implementation, it is not difficult to re-label the decision variables using $\lceil m \log 3 \rceil$ binary variables.

Algorithm 2 Schöning Walk & Oracle

```
function ORACLE($x_0, w$)
    return TRUE if SCHOENINGWALK($x_0, w$) satisfies all clauses, else FALSE
end function

function SCHOENINGWALK($x, w$)
    for $j = 1 \ldots m$ do
        if $x$ violates one of $C_1, \ldots, C_L$ then
            $k \leftarrow$ index of first unsatisfied clause
            $l \leftarrow$ index of the $w_j$-th variable occurring in $C_k$
            $x \leftarrow x$, with the $l$-th bit of $x$ flipped
        end if
    end for
    return $x$
end function
```

For the different variants of partial Groverizations discussed below, we will fix a subset of arguments to the oracle, and consider it as a function of the remaining ones. Fixed arguments will be denoted as subscripts, e.g. $\textsc{Oracle}_w : x \mapsto \textsc{Oracle}(x, w)$. With these conventions, we have:

Algorithm 3 Groverized Initialization

```
for $i = 1...N_2$ do
    $w \leftarrow$ uniformly random value from $\{1, 2, 3\}^{\times m}$
    $x \leftarrow$ Grover-search for $\lfloor \sqrt{N_1} \rfloor$ iterations using $\textsc{Oracle}_w()$
    if $x$ satisfies all clauses then
        return $x$
    end if
end for
```

Algorithm 4 Groverized Walk

```
for $i = 1...N_1$ do
    $x_0 \leftarrow$ uniformly random value from $\{0, 1\}^{\times n}$
    $w \leftarrow$ Grover-search for $\lfloor \sqrt{N_2} \rfloor$ iterations using $\textsc{Oracle}_{x_0}()$
    $x \leftarrow \textsc{SchoeningWalk}(x_0, w)$
    if $x$ satisfies all clauses then
        return $x$
    end if
end for
return False
```

One may note that the Grover search in the Groverized walk only is guaranteed to succeed (with high probability) for a specific collection of initial states. The number of rounds N_1 of the outer loop is selected in such a way that it with high probability hits the set of advantageous initial states at least once, thus allowing the Grover-procedure to reach the satisfying assignment. Similar remarks apply to the other partial Groverizations.

Next, we discuss the "fractional searches". In the first one, the argument x of the oracle is broken up as $x = (x_c, x_q)$ with x_q taking $\lfloor z \cdot n \rfloor$ bits and x_c being $\lceil (1 - z) \cdot n \rceil$ bits long. Here, $z \in [0, 1]$ is a free parameter whose value will be determined below.

Algorithm 5 Fractional Groverized Initialization

```
for $i = 1...N_2$ do
    $w \leftarrow$ uniformly random value from $\{1, 2, 3\}^{\times m}$
    for $j = 1...N_1^{(c)}$ do
        $x_c \leftarrow$ uniformly random value from $\{0, 1\}^{\times \lceil (1-z)n \rceil}$
        $x_q \leftarrow$ Grover-search for $\left\lfloor \sqrt{N_1^{(q)}} \right\rfloor$ iterations using $\textsc{Oracle}_{(x_c,w)}()$
        $x = (x_c, x_q)$
        if $x$ satisfies all clauses then
            return $x$
        end if
    end for
end for
return False
```

The second fractional algorithm breaks up the walk randomness as $w = (w_c, w_q)$ with $w_c \in \{1, 2, 3\}^{m_c}$ and $w_q \in \{1, 2, 3\}^{m_q}$ respectively. Again, the values of m_c, m_q are chosen later.

Algorithm 6 Fractional Groverized Walk

```
for $i = 1...N_1$ do
    $x_0 \leftarrow$ uniformly random value from $\{0,1\}^{\times n}$
    for $j = 1...N_2^{(c)}$ do
        $w_c \leftarrow$ uniformly random value from $\{1,2,3\}^{\times m_c}$
        $w_q \leftarrow$ Grover-search for $\left\lfloor\sqrt{N_2^{(q)}}\right\rfloor$ iterations using ORACLE$_{(x_0,w_c)}()$
        $w = (w_c, w_q)$
        $x \leftarrow$ SCHOENINGWALK$(x_0, w)$
        if $x$ satisfies all clauses then
            return $x$
        end if
    end for
end for
return False
```

In the final algorithm, a fraction of $z \in [0,1]$ of both types of variables, the ones corresponding to the initialization and the ones corresponding to the walk, will be treated quantum mechanically.

Algorithm 7 Evenly Fractionalized Grover

```
for $i = 1...N^{(c)}$ do
    $x_c \leftarrow$ uniformly random value from $\{0,1\}^{\times\lceil(1-z)n\rceil}$
    $w_c \leftarrow$ uniformly random value from $\{1,2,3\}^{\times\lceil(1-z)m\rceil}$
    $(x_q, w_q) \leftarrow$ Grover-search for $\left\lfloor\sqrt{N^{(q)}}\right\rfloor$ iterations using ORACLE$_{(x_c,w_c)}()$
    $w = (w_c, w_q)$
    $x_0 = (x_c, x_q)$
    $x \leftarrow$ SCHOENINGWALK$(x_0, w)$
    if $x$ satisfies all clauses then
        return $x$
    end if
end for
return False
```

IV. RUN-TIME ANALYSIS

We will now lower-bound the probability of success of the various approaches. As a preparation, in Sec. IV A, we give a brief account of the analysis of the classical case, before moving on to the Groverized versions in Sec. IV B.

A. The classical Schöning process

The main ideas of the classical analysis are close to their presentation in Refs. [1, 2]. We work in the Markovian model $(d_l)_l$ for the behavior of the Hamming distances, as laid out in Sec. II B. Frequently, it will be convenient to measure quantities "in units of n or m". For example, we will soon choose a number $\kappa \in [0,1]$ and assume that the initial value d_0 is equal to κn. Of course, this only makes sense if κn is an integer. In order to keep the notation clean, we will implicitly assume that such expressions have been rounded to the next integer.

Choose numbers $\kappa, \nu \in [0,1]$. A given walk $(d_l)_l$ is certainly successful (in the sense that $d_m \leq 0$) if

1. The initial value is $d_0 = \kappa n$,

2. the random walk decreases the Hamming distance in exactly νm of its m steps, and

3. the condition

$$\kappa n \leq (2\nu - 1)m \tag{6}$$

holds.

Indeed, the right hand side of (6) is the difference between the number of steps where the Hamming distance has been decreased, νm, and the number of steps where the Hamming distance has been increased, $(1-\nu)m$.

For any fixed pair of values κ, ν subject to (6), we will now compute the probability of this particular route to success. Denote the first event by E_1 and the second event by E_2. They occur with respective probabilities

$$P(E_1) = \frac{1}{2^n}\binom{n}{\kappa n}, \qquad P(E_2) = \binom{m}{\nu m}\left(\frac{1}{3}\right)^{\nu m}\left(\frac{2}{3}\right)^{(1-\nu)m}. \tag{7}$$

Since the two events are independent, the success probability of the walk is lower-bounded by

$$P(x_m = x^\star|\kappa) \geq P(d_m \leq 0|x) \geq P(E_1 \wedge E_2) = P(E_1)P(E_2) = \frac{1}{2^n}\binom{n}{\kappa n}\binom{m}{\nu m}\left(\frac{1}{3}\right)^{\nu m}\left(\frac{2}{3}\right)^{(1-\nu)m}. \tag{8}$$

The various binomial coefficients can be conveniently related to entropies. To this end, recall the definition of the *binary entropy function*

$$H(p) = -p\log p - (1-p)\log(1-p) \quad \text{for} \quad p \in [0,1],$$

and the *relative entropy*

$$D(p \,\|\, q) = -p\log q - (1-p)\log(1-q) - H(p) \quad \text{for} \quad p, q \in [0,1].$$

Then using the well-known estimate [8, Chapter 11.1]

$$\frac{1}{n+1}2^{nH(\kappa)} \leq \binom{n}{\kappa n} \leq 2^{nH(\kappa)},$$

Equation (8) can, after some straight-forward calculations, be concisely rewritten as

$$P(d_m \leq 0|x) \gtrsim 2^{-(1-H(\kappa))n}2^{-D(\nu\|\frac{1}{3})m}, \tag{9}$$

where $\gtrsim$ denotes an inequality holds asymptotically, up to a polynomial factor. Equation (9) directly gives an upper bound on the rate γ defined in (5). Since the rate expresses the logarithm of the complexity “in units of n”, it makes sense to also express the length of the walk in terms of $\mu := m/n$. Then:

$$\gamma = -\lim_{n\to\infty}\frac{1}{n}\log P(d_m \leq 0|x) \leq 1 - H(\kappa) + \mu D(\nu \,\|\, 1/3) =: \gamma(\mu,\kappa,\nu). \tag{10}$$

In particular, the infimum of $\gamma(\mu,\kappa,\nu)$ subject to the constraints (6) and $0 \leq \mu, 0 \leq \nu, \kappa \leq 1$ is a valid bound for γ. We will perform such optimizations explicitly for the partially Groverized versions in Sec. IV B. For the classical procedure, we just state the final result:

$$\mu = 1, \qquad \kappa = \frac{1}{3}, \qquad \nu = \frac{2}{3}, \qquad \gamma_C = \log\frac{4}{3} \simeq 0.4150. \tag{11}$$

Remark: One might be tempted to search a tighter bound by summing the contributions to the probability of success that arise from all consistent values for μ, κ, ν, instead of just considering the extremal value. However, the rate of a sum of exponentially processes is asymptotically determined by the rate of the dominating summand alone, i.e. for all collections of $\gamma_i > 0$, it holds that

$$\lim_{n\to\infty} -\frac{1}{n}\log\sum_i 2^{-\gamma_i n} = \sup_i \gamma_i$$

(assuming convergence). Therefore, considering only the dominating term does not affect the overall asymptotic rate.

B. Partially Groverized processes

In this section, we derive the main results of this paper: Bounds on the asymptotic rates for partially Groverized versions of Schöning’s scheme.

For the parameters N_1, N_2, we choose constant multiples of $1/P(E_1), 1/P(E_2)$ respectively. The value of the constant depends on the acceptable probability ϵ of failure, as exhibited in Eq. (4). Since this constant does not effect the rate, we will not specify it here. The probabilities do depend essentially on the parameters μ, κ, ν, though. We will therefore write $N_1(\kappa)$ and $N_2(\mu, \nu)$. Because the asymptotic complexity of a Grover search is the square root of the classical complexity, the rate function of GI is then given by

$$\gamma_{\text{GI}}(\mu, \kappa, \nu) = \lim_{n\to\infty} \frac{1}{n} \log \left(\sqrt{N_1(\kappa) N_2(\nu, \mu)} \right) = \frac{1 - H(\kappa)}{2} + \mu D(\nu \parallel 1/3). \tag{12}$$

Likewise, the required coherence time scales with the number of Grover iterations, i.e. as $O^*(2^{\chi n})$, for

$$\chi(\kappa) := \lim_{n\to\infty} \frac{1}{n} \log \sqrt{N_1(\kappa)} = \frac{1 - H(\kappa)}{2}. \tag{13}$$

The parameters are constrained by

$$0 \le \kappa \le 1, \quad 0 \le \mu, \quad 0 \le \nu \le 1, \quad \frac{\kappa}{2\nu - 1} \le \mu, \tag{14}$$

where the final condition is a re-arranged version of the success criterion (6).

We now determine the minimal rate γ_{GI} over the consistent parameters. Because relative entropy is non-negative, it is always advantageous to reduce the value of μ until it is minimal subject to the constraints. This is achieved by changing the final inequality in (14) to equality. Re-arranging, we arrive at

$$0 \le \kappa \le 1, \quad 0 \le \mu, \quad \nu = \frac{1}{2} + \frac{\kappa}{2\mu}, \tag{15}$$

which allows us to eliminate $\nu = \nu(\kappa, \mu)$ from the problem. Varying γ with respect to μ gives rise to the criticality condition

$$0 \overset{!}{=} \partial_\mu \gamma_{\text{GI}}(\kappa, \mu) = \partial_\mu \, \mu D(1/2 + \kappa/(2\mu) \parallel 1/3) = \frac{1}{2} \log \left(\frac{\mu + \kappa}{\mu} \right) + \frac{1}{2} \log \left(\frac{\mu - \kappa}{\mu} \right) + \log 3 - \frac{3}{2}. \tag{16}$$

This can be solved explicitly e.g. using a computer algebra system [9], leading to

$$\mu = 3\kappa \qquad \Rightarrow \qquad \nu = \frac{2}{3}, \quad \mu D(\nu \parallel 1/3) = \kappa. \tag{17}$$

Eliminating μ, we get

$$\begin{aligned} \gamma_{\text{GI}}(\kappa) &= \frac{1 - H(\kappa)}{2} + \kappa, \\ \chi_{\text{GI}}(\kappa) &= \frac{1 - H(\kappa)}{2}. \end{aligned} \tag{18}$$

The pair of equations (18) contain all information about the asymptotic behavior of the Groverized Initialization procedure. Each value of κ gives a solution for the two undetermined constants $N_1(\kappa), N_2(\mu = 3\kappa, \nu = \frac{2}{3})$ in Alg. 3, in such a way that it will run with a small probability of returning a false negative. Varying κ, we thus obtain a family of algorithms that find different compromises between the required coherence time and the total runtime. The achievable pairs of values are shown in Fig. 2.

Finally, we explicitly determine the minimal rate achievable in the Groverized Initialization scheme. With the help of a computer algebra system [9], one easily finds

$$0 \overset{!}{=} \partial_\kappa \gamma_{\text{GI}}(\kappa) = \frac{1}{2} \log \frac{\kappa}{1 - \kappa} + 1 \quad \Leftrightarrow \quad \log \left(\frac{1}{\kappa} - 1 \right) = 2 \quad \Rightarrow \quad \kappa = \frac{1}{5} \tag{19}$$

which gives

$$\mu = \frac{3}{5}, \qquad \gamma_{\text{GI}} = \frac{3 - \log 5}{2} \approx 0.339, \qquad \chi_{\text{GI}} \simeq 0.139. \tag{20}$$

Remark: One can cast the final minimization into the form

$$\gamma_{\text{GI}} = \inf_\kappa \gamma_{\text{GI}}(\kappa) = \inf_\kappa \left(\frac{1 - H(\kappa)}{2} + \kappa \right) = -\sup_\kappa \left(-\kappa - \frac{H(\kappa) - 1}{2} \right).$$

This expression shows that the optimization amounts to computing a Legendre transform. Indeed, with $f(\kappa) := 1/2(H(\kappa) - 1)$, the right hand side equals $-f^*(-1)$. For physicist readers, it might be amusing to note that $S(n\kappa) = nH(n\kappa)$ formally equals the entropy of an n-spin paramagnet as a function of the total magnetization. The Legendre transform of the entropy is a Massieu thermodynamic potential, equal to F/T (with F the free energy) expressed as a function of the inverse temperature [10, Chapter 5.4]. We will, however, not pursue this analogy here.

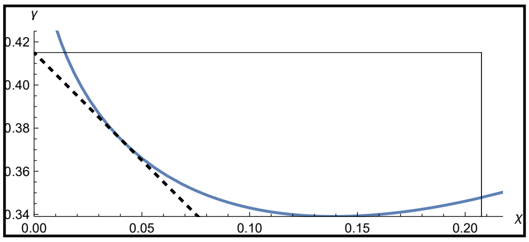

FIG. 2: Rates $(\chi_{\rm GI}(\kappa), \gamma_{\rm GI}(\kappa))$ for the required coherence time and the total runtime of the Groverized Initialization algorithm, as the parameter κ is varied. The horizontal bar denotes the runtime rate achieved by the classical Schöning process. In other words, points above this line are uninteresting. The vertical bar denotes the coherence rate that allows one to run a completely Groverized version of the Schöning process. This, arguably, makes points to the right of this line uninteresting as well. Points to the left of the minimum (at $(\gamma, \chi) \simeq (0.339, 0.139)$) can represent advantageous choices if either the total coherence time of a quantum computer is limited, or a larger degree of parallelization is desired. The dashed line is the lower bound on the runtime rate given the coherence time, as introduced in Fig. 1. It is achieved for $\kappa = \frac{1}{3}$.

2. *Groverized Walk, Algorithm 4*

The analysis proceeds in close analogy to the above case. The asymptotic rate function of GW is

$$\gamma_{\rm GW}(\kappa,\mu,\nu) = \lim_{n\to\infty} \frac{1}{n} \log\left(N_1(\kappa)\sqrt{N_2(\nu,\mu)}\right) = 1 - H(\kappa) + \frac{\mu}{2} D(\nu \,\|\, 1/3), \tag{21}$$

subject to the set of constraints (14). The parameters ν, μ can be treated in exactly the same way as before, leading again to (17). In particular, the coherence time rate takes the simple form $\chi = \kappa/2$, which allows us to eliminate κ in favor of χ. We immediately obtain

$$\gamma_{\rm GW}(\chi) = 1 - H(2\chi) + \chi. \tag{22}$$

Again, it is not difficult to solve for the lowest runtime [9]:

$$\mu = 3(\sqrt{2}-1), \qquad \kappa = \sqrt{2}-1, \qquad \gamma_{\rm GW} \approx 0.228, \qquad \chi_{\rm GW} \simeq 0.2071. \tag{23}$$

At the optimal point, the runtime scales with a rate that is very close to the one of a full Groverization of Schöning's process, namely $\gamma_{\rm FG} = \gamma_{\rm C}/2 \simeq .2075$. The flip side is that the required coherence times are basically identical:

$$\chi_{\rm FG} - \chi_{\rm GW} \simeq 0.0004.$$

The findings are summarized in Fig. 3.

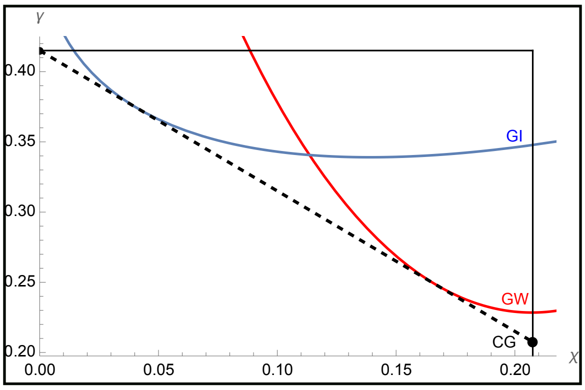

FIG. 3: The runtime rate vs coherence time rate curves for Groverized Initialization (GI, blue) and Groverized Walk (GW, red). The point marked "CG" at the bottom right of the diagram represents the complete Groverization of the Schöning process. For long coherence times, GW is preferable, while for shorter coherence times GI achieves a lower total runtime.

3. *Fractional Groverized Initialization, Algorithm 5*

In the case of Alg. 5, the initial Hamming distance is the sum of two terms $d_0 = \kappa_c(1-z)n + \kappa_q zn$, which model $d_H(x_c, x_c^\star)$ and $d_H(x_q, x_q^\star)$ respectively. Define the analogues

$$P(E_1^c) = \frac{1}{2^{(1-z)n}} \binom{(1-z)n}{\kappa_c(1-z)n}, \qquad P(E_1^q) = \frac{1}{2^{zn}} \binom{zn}{\kappa_q zn}$$

of $P(E_1)$ introduced in Eq. (7). Analogous to the discussion in Sec. IV B 1, the parameters N_1^c, N_1^q are defined as the reciprocals of these probabilities, times a constant that influences the probability of a false negative, but will not be discussed as it has no impact on the asymptotic rates. The success criterion is now

$$(1-z)\kappa_c + z\kappa_q \leq (2\nu - 1)\mu$$

and the other constraints are

$$0 \leq \kappa_c, \kappa_q, \nu, z \leq 1, \qquad 0 \leq \mu.$$

The asymptotic rate function for the runtime of FGI reads

$$\begin{aligned} \gamma_{\text{FGI}}(\kappa_c, \kappa_q, \nu, \mu; z) &= \lim_{n\to\infty} \frac{1}{n} \log\left(N_1^c(\kappa_c; z)\sqrt{N_1^q(\kappa_q; z)}N_2(\nu, \mu)\right) \\ &= (1-z)\left(1 - H(\kappa_c)\right) + \frac{z}{2}\left(1 - H(\kappa_q)\right) + \mu D(\nu \,\|\, 1/3) \end{aligned} \tag{24}$$

Arguing as in Sec. IV B 1, the inequality in the success criterion may be replaced by an equality. Solving for ν gives

$$\nu = \frac{1}{2} + \frac{(1-z)\kappa_c + z\kappa_q}{2\mu}.$$

We proceed as in the first two cases. Criticality of $\partial_\mu \gamma_{\text{FGI}}$ with respect to μ occurs at

$$\mu = 3((1-z)\kappa_c + z\kappa_q) \qquad \Rightarrow \qquad \mu D(\nu \,\|\, 1/3) = (1-z)\kappa_c + z\kappa_q, \quad \nu = \frac{2}{3}.$$

Plugging in, we arrive at

$$\gamma_{\text{FGI}}(\kappa_c, \kappa_q; z) = (1-z)\left(1 - H(\kappa_c) + \kappa_c\right) + z\left(\frac{1 - H(\kappa_q)}{2} + \kappa_q\right). \tag{25}$$

In other words, the runtime rate function is a convex combination of the ones for the classical Schöning process and for the GI scheme, with weights $(1-z), z$ respectively. Because the classical part does not affect the coherence time, we may set κ_c to its optimal value $\kappa_c^* = 1/3$ (c.f. Eq. (11)). Geometrically, as we vary $z \in [0, 1]$, Eq. (25) describes a line connection $(\chi_{\text{GI}}(\kappa_q), \gamma_{\text{GI}}(\kappa_q))$ with the parameters of the classical Schöning process $(0, \gamma_{\text{C}})$. By the convexity of the GI curve, the fractional algorithm will have a better runtime rate to the left of the value of κ_q at which the line becomes tangent to the curve. In other words, the critical κ_q is defined by the condition

$$\frac{\partial \gamma_{\text{GI}}}{\partial \chi} = \frac{\gamma_{\text{GI}} - \gamma_{\text{C}}}{\chi}.$$

By a computer calculation [9], this happens for $\kappa_q = \frac{1}{3}$ (i.e. equal to κ_c), resulting in the following curve:

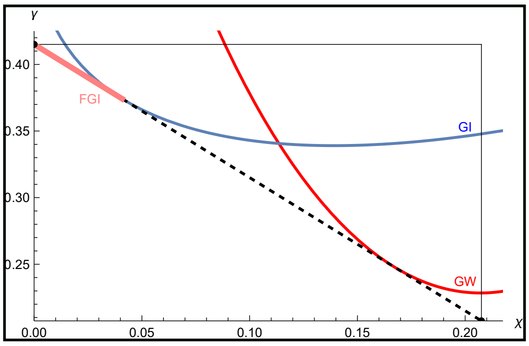

FIG. 4: The runtime rate vs coherence time rate for the FGI algorithm. This fractional scheme's performance is the convex combination of the classical point $(0, \gamma_C)$, and GI at the tangent point to the theoretical lower bound. One can note that the FGI partially saturates the optimal performance relation.

4. Fractional Groverized Walk, Algorithm 6

In the FGW scheme, we assume that the classical and Groverized walks decrease the Hamming distance in exactly $\nu_c m_c$ and $\nu_q m_q$ steps, respectively, where we have used a subscript to differentiate between the classical and Groverized random walks. The probabilities of such walks occurring is given by:

$$P(E_2^c) = \binom{m_c}{\nu_c m_c} \left(\frac{1}{3}\right)^{\nu_c m_c} \left(\frac{2}{3}\right)^{(1-\nu_c)m_c}, \qquad P(E_2^q) = \binom{m_q}{\nu_q m_q} \left(\frac{1}{3}\right)^{\nu_q m_q} \left(\frac{2}{3}\right)^{(1-\nu_q)m_q} \tag{26}$$

Analogous to the discussion in Sec. IV B 1, the parameters N_2^c, N_2^q are defined as the reciprocals of the probabilities $P(E_2^c), P(E_2^q)$, times a constant that influences the probability failure, but will not be discussed as it has no impact on the asymptotic rates. We further parameterize the walk lengths as $m_c = \mu_c n$ and $m_q = \mu_q n$. The runtime rate is

$$\begin{aligned} \gamma_{\text{FGW}}(\kappa, \nu_c, \mu_c, \nu_q, \mu_q) &= \lim_{n\to\infty} \frac{1}{n} \log \left(N_1(\kappa) N_2^c(\nu_c, \mu_c) \sqrt{N_2^q(\nu_q, \mu_q)} \right) \\ &= 1 - H(\kappa) + \mu_c D(\nu_c \parallel 1/3) + \frac{\mu_q}{2} D(\nu_q \parallel 1/3) \end{aligned} \tag{27}$$

with parameters subject to the constraints

$$\begin{aligned} &0 \le \kappa \le 1, \\ &0 \le \mu_c, \mu_q, \\ &0 \le \nu_c, \nu_q \le 1, \\ &\kappa \le (2\nu_c - 1)\mu_c + (2\nu_q - 1)\mu_q. \end{aligned} \tag{28}$$

The first steps of the analysis should now be familiar. There is no loss of generality in assuming that the final inequality is tight, which can be re-arranged to give

$$\nu_q = \frac{\kappa - (2\nu_c - 1)\mu_c}{2\mu_q} + \frac{1}{2}.$$

The rate γ_{FGW} is stationary as a function of μ_q if

$$\mu_q = 3(\kappa - (2\nu_c - 1)\mu_c) \qquad \Rightarrow \qquad \nu_q = \frac{2}{3}, \frac{\mu_q}{2} D(\nu_q \parallel 1/3) = 1/2(\kappa - (2\nu_c - 1)\mu_c) = \chi(\kappa, \nu_c, \mu_c).$$

Eliminating κ in favor of the coherence rate χ gives

$$\kappa = 2\chi + (2\nu_c - 1)\mu_c$$

and thus

$$\mu_q = 6\chi, \qquad \gamma_{\mathrm{FGW}}(\nu_c, \mu_c; \chi) = 1 - H(2\chi + (2\nu_c - 1)\mu_c) + \mu_c D(\nu_c \parallel 1/3) + \chi.$$

We now need to minimize γ_{FGW} for fixed χ as a function of μ_c, ν_c, subject to

$$\begin{aligned} &0 \le 2\chi + (2\nu_c - 1)\mu_c \le 1, \\ &0 \le \mu_c, \\ &0 \le \nu_c \le 1. \end{aligned}$$

We may assume that $\mu_c \neq 0$, for else we are just replicating the GW scheme. A computer calculation [9] gives

$$\partial_{\mu_c}(\gamma \ln 2) + \frac{2 - 4\nu_c}{4\mu_c}\partial_{\nu_c}(\gamma \ln 2) = -\arctan(1 - 2\nu_c) + \ln(3 - 3\nu_c) - \frac{1}{2}\ln 2,$$

which has zeros at $\nu_c = \frac{1}{3}$ and $\nu_c = \frac{2}{3}$.

For $\nu_c = \frac{1}{3}$, one finds

$$\partial_{\mu_c}(\gamma \ln 2) = \frac{2}{3}\arctan(1 - 4\chi + 2/3\mu_c)$$

which has one zero, at $\mu_c = \frac{3}{2}(4\chi - 1)$. The constraint $\mu_c \geq 0$ then implies $\chi \geq \frac{1}{4}$. But this is larger than the coherence time rate $\gamma_{\mathrm{C}}/2 \simeq 0.208$ sufficient to implement a completely Groverized version of Schöning's process, so this solution is not of interest.

We turn to the other solution, $\nu_c = \frac{2}{3}$. For it,

$$\partial_{\mu_c}(\gamma \ln 2) = 1/3(-2\,\mathrm{arctanh}(1 - 4\chi - (2\mu_c)/3) + \ln(2)),$$

which has one zero:

$$\mu_c = 1 - 6\chi \qquad \Rightarrow \qquad \mu_q = 6\chi, \quad \nu_c = \nu_q = \frac{2}{3}, \quad \gamma_{\mathrm{FGW}} = \gamma_{\mathrm{C}} - \chi.$$

The runtime vs coherence rate curve for the FGW scheme is given in the following figure:

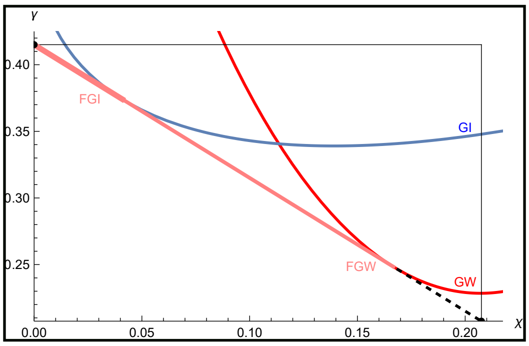

FIG. 5: The runtime rate vs coherence time rate for the FGW algorithm. This fractional scheme's performance connects the GW curve to the classical Schöning point and is tangent to the curve. It achieves the optimal performance relation partially for a larger regime than FGI and for low coherence times, it comes to lie on top of the FGI line.

5. *Evenly Fractionalized Grover*

The runtime rate is

$$\gamma_{\mathrm{EFG}} = (1 - z)\big(1 - H(\kappa_c) + \mu_c D(\nu_c \parallel 1/3)\big) + z/2\big(1 - H(\kappa_q) + \mu_q D(\nu_q \parallel 1/3)\big) \tag{29}$$

with success criterion

$$(1 - z)\kappa_c + z\kappa_q = (1 - z)(2\nu_c - 1)\mu_c + z(2\nu_q - 1)\mu_q,$$

which is in particular true if the following two equations hold

$$\kappa_c = (2\nu_c - 1)\mu_c, \qquad \kappa_q = (2\nu_q - 1)\mu_q.$$

But this is just the convex interpolation between a completely classical and a completely Groverized process. In particular, by choosing the parameters as for the original Schöning process

$$\nu_c = \nu_q = \frac{2}{3}, \quad \kappa_c = \kappa_q = \frac{1}{3}, \quad \mu_c = \mu_q = 1,$$

we obtain a coherence time–runtime rate curve that linearly connects the classical point $(0, \gamma_C)$ to the completely Groverized one $(\gamma_C/2, \gamma_C/2)$ (Fig. 6).

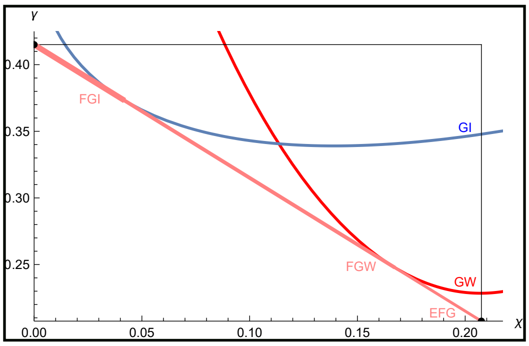

FIG. 6: Runtime–coherence time rate curves for the covered algorithms. The linear interpolation between the classical and the completely Groverized points are realizable using an increasing number of methods – first only EFG, then also FGW, finally also FGI – as the coherence time decreases.

C. A heuristic de-randomization of the GI schemes

In this section, we provide evidence that the Groverized initialization schemes can reach further into the γ–χ chart than what the Markovian model suggests. To see why this is plausible, note that the role of randomness for the initial configuration x is very different from the role of randomness for the walk decisions w. In the first case, there is an “absolute measures of the quality of the initial configuration”, namely the Hamming distance to the solution. The probability that the walk does find the solution is quite obviously a function of that metric. Therefore, baring major algorithmic insights, it is unavoidable to consider many different initial configurations before encountering one that will likely lead to a solution.

In contrast, it is not implausible that “every walk works for equally many initial configurations”, i.e. that there are no choices for w that are “intrinsically better than others”. More precisely, it seems reasonable to assume that for sufficiently large n, and generic SAT formulas, it holds that with high probability in w

$$\begin{aligned} &-\frac{1}{n}\log\Big(\Pr_x[\textsc{SchoeningWalk}(x,w) = x^\star \,|\, d_H(x,x^\star) = h, w]\Big) \qquad (30)\\ &\simeq -\frac{1}{n}\log\Big(\Pr_{x,w'}[\textsc{SchoeningWalk}(x,w') = x^\star \,|\, d_H(x,x^\star) = h]\Big). \end{aligned}$$

The right hand side can be easily calculated, as by Ref. [1], for $\mu = 3$,

$$\Pr_{x,w}[\textsc{SchoeningWalk}(x,w) = x^\star \,|\, d_H(x,x^\star) = h] = 2^{-h}.$$

Under Assumption (30), one can restrict the outer loop over w’s from Alg. 3 to $N_2 = 1$ iteration, and compensate by increasing the number of Grover iterations for x to $N_1 = O^*(2^{\gamma_C/2n})$. In other words, the Groverized Initialization scheme with these parameters would lie on the optimal point $(\chi, \gamma) = (\gamma_C/2, \gamma_C/2)$.

Being even bolder, one could then speculate that the analysis of Sec. IV B 3 carries over and that, as one varies the fraction of initialization bits that are subjected to a Grover search, one could trace out the optimal (χ, γ)-line. In other words, it does not seem impossible that the following Alg. 8, with parameter choice

$$N_1^{(c)} = O^*(2^{\gamma_C(1-z)n}), \qquad N_1^{(q)} = O^*(2^{\gamma_C zn/2}),$$

achieves the optimal trade-off.

Algorithm 8 Heuristically De-Randomized Fractional Groverized Initialization

```
$w \leftarrow$ uniformly random value from $\{1,2,3\}^{\times m}$
for $j = 1 \ldots N_1^{(c)}$ do
    $x_c \leftarrow$ uniformly random value from $\{0,1\}^{\times \lceil (1-z)n \rceil}$
    $x_q \leftarrow$ Grover-search for $\lfloor \sqrt{N_1^{(q)}} \rfloor$ iterations using $\textsc{Oracle}_{(x_c,w)}()$
    $x = (x_c, x_q)$
    if $x$ satisfies all clauses then
        return $x$
    end if
end for
return False
```

To gather evidence in favor of Assumption (30), we have resorted to numerical methods. A first ansatz is to compute the l.h.s. of Eq. (30) exactly, which is possible for small values of n by iterating over all 2^n assignments to x. Results are shown in Fig. 7 for a randomly chosen set of 3-SAT formulas with $n = 20$ variables, $L = 91$ clauses. The number of satisfying assignments t_0 of the formulas are varied. Only the case $t_0 = 1$ can be directly compared to the analytic bounds. However, note that even for this case, the empirically observed rate of $\gamma_{\mathrm{GI}} \simeq .12 \pm .02$ is much lower than the value $\gamma_{\mathrm{C}}/2 \simeq .208$ that we would expect theoretically. Presumably, $n = 20$ is still too small to show the asymptotic behavior.

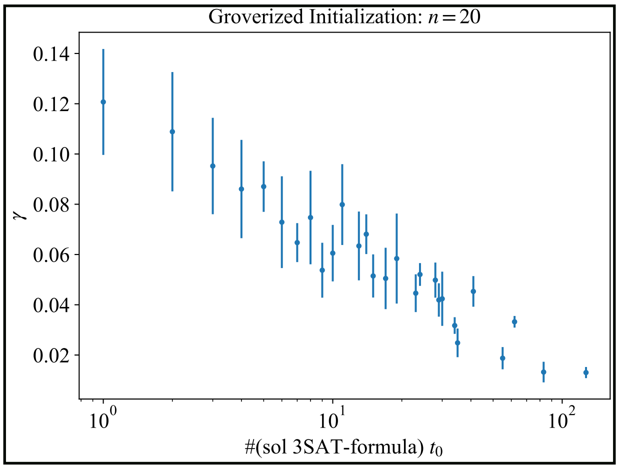

FIG. 7: Plot of the runtime rate for the heuristically de-randomzied GI scheme. Error bars indicate variation as a function of the formulas and the walk variables w. On the x-axis, we show the number of satisfying assignments in the formula. Only the case of $t_0 = 1$ should be directly comparable to the analytic bounds. The empirically observed behavior is much better than the analytic results, suggesting that $n = 20$ is too small to capture the asymptotic behavior.

To test this assumption, we had to turn to numerical heuristics, to at least probe the behavior for much larger values of n, where an exact computation is no longer possible. The results are shown in Fig. 8. We used a SAT instance with $n = 1414$ variables that we believe to have a single satisfying assignment $x^\star$ which is explicitly known. To generate the instance, a 128-bit plain text was encoded by a 128-bit key using the XTEA block cipher truncated to three rounds. The formula represents the conditions on an input key to map the known plain text to the known ciphertext. The clauses are designed such that they enforce the correct evaluation of bit-wise operations of the algorithm with respect to the given input and output. XTEA was restricted to three rounds in order to keep the size of the formula manageable. While we have no formal proof, it is reasonable to assume that there is a unique key that satisfies the formula. This is supported by consistency checks in terms of running SAT solvers on a version of this problems with even fewer rounds [11].

Let us denote the sphere of strings with Hamming distance h from $x^\star$ by $M^h(x^\star)$. For a fixed walk randomness w, and for $h = 1, \ldots 11$, we have drawn x uniformly from $M^h(x^\star)$. In order to compare the numerical results to the theory prediction, we have to use the value of the right hand side of Assumption (30) for non-asymptotic values of n. The following plot shows the empirically estimated probabilities of Schöning's walk (with $\mu = 3$) arriving at the solution, when starting from a random initial configuration of given Hamming distance. The findings show the expected behavior of averaging over w, already for a fixed random value of w. In this sense, they are compatible with Assumption (30). We note, however, that we were not able to probe the assumption for larger values of h. Garnering a better understanding for the concentration properties of the Schöning walk as a function of the walk choices remains therefore an open question.

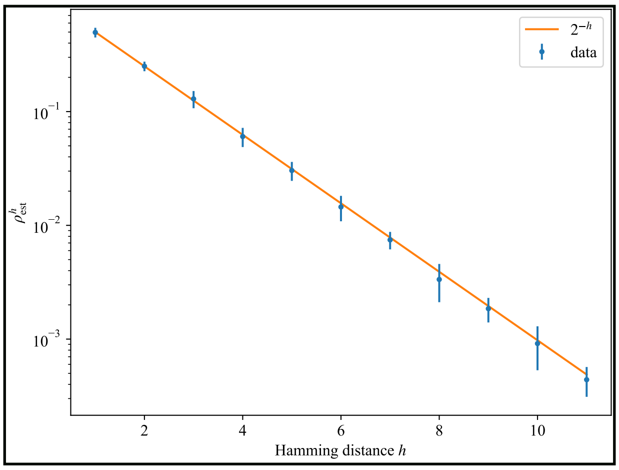

FIG. 8: Estimated probability for a uniformly random initial configuration x with Hamming distance h to be mapped to $x^\star$ under a Schöning walk, for a fixed, randomly chosen set of walk decisions w (c.f. Alg. 8). The SAT instance has $n = 1414$ variables and is believed to have a unique satisfying assignment [11]. For each data point, 10^4 initial configurations x, were sampled uniformly from the Hamming distance sphere $M^h(x^\star)$. The results agree well with the theoretical prediction under Assumption (30) (orange line).

V. CIRCUITS

In this section, we discuss an implementation of the partial Groverization schemes and present the main building blocks of their quantum circuits. Given n variables and the length of Schöning's walk m, the quantum implementation requires $n+m\log 3$ qubits to encode the initializations and walk randomness. The oracles of the partial Groverization schemes are some adaptation of one or more Schöning walks, and regardless of the search space they act on, the label of the violated clause at each step needs to be stored in their workspaces. This is necessary since such oracles are typically realized using uncomputation, therefore, $\log L$ extra auxiliary qubits are needed at each step, amounting to $m \log L$ qubits in total for the workspace. As a result, encoding any Groverization of Schöning's algorithm asymptotically needs $n + (\log 3 + \log L)m$ qubits.

Figure 9 represents a single step of Schöning walk, schematically. The first register encodes the space of all possible initialization. The gates ev_j, for $j \in \{1, .., L\}$, act on the first two registers. Each gate consists of a few controlled-gates where the control qubits correspond to the three variables in the j-th clause, and the target qubit is the second register. The second register is an auxiliary qubit, initially set to $|0\rangle$, and is negated as soon as the first violated clause is detected. The third register consists of $\log L$ auxiliary qubits that are used to count the number of clauses from where the first violated clause has happened. The last register is a qutrit providing the randomness of the corresponding walk step. The controlled-gates ch_j, for $j \in \{1, .., L\}$ act on the first three registers, and take care of variable flipping wherever the first violated clause is detected. The $0 \vee 1 \vee 2$ block represents a triple controlled-gate where the control qutrit is the subspaces corresponding to the computational basis states $|0\rangle, |1\rangle, |2\rangle$. Figure 10 depicts the controlled-gates including ch_j, in detail. The sub-figure on the right shows the corresponding controlled-gate for GI, where the walk randomness is fed classically to the last register.

All partial Groverization of Schöning algorithm can be implemented using slight modifications. For the GW algorithm, the n-qubit variable register will not be initialized in the uniform superposition of all possible assignments $|+\rangle^{\otimes n}$, but rather in a state with classically randomly defined variables $|x_1 \cdots x_n\rangle$. For the GI algorithm the qutrit within every Schöning's step can be removed since we can, for every Schöning's step, generate a random number $r \in \{0, 1, 2\}$ and apply only the X gates based on the classically determined r (see figure 10).

VI. SUMMARY & OUTLOOK

This work considers hybrid schemes for search-based quantum algorithms, with the aim to allow for parallelizability, and to reduce the need for long coherence times. The basic gist is to partition the randomness of an underlying classical probabilistic algorithm into a part that is subject to Grover search, while the rest is sampled classically. Such 'partial Groverizations' allow for parallelization of the classical sampling, as well as enable adaption to available coherence times. We consider exponential-time

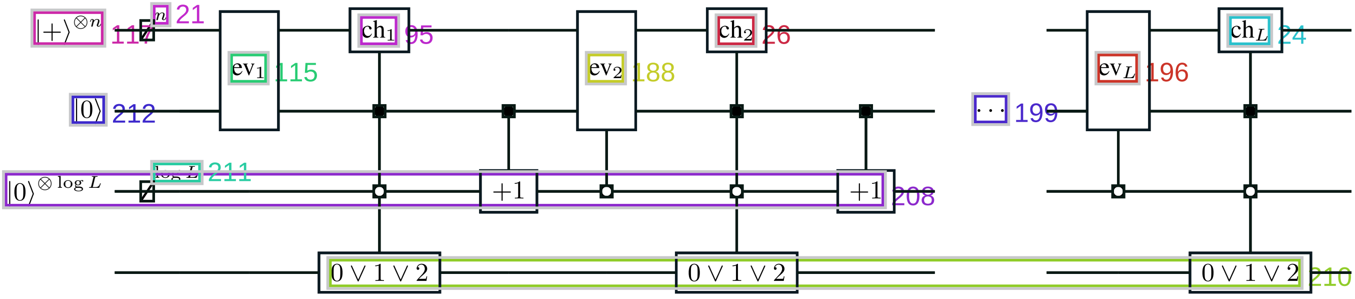

FIG. 9: The quantum implementation of a single Schöning's step for a general implementation of the partial Groverization of Schöning's algorithm. The ev_j gates evaluate the j-th clause on the corresponding variables and the controlled-gates containing ch_i and $0 \vee 1 \vee 2$ act on all the registers and check if the j-th clause is the first violated clause and if so, flip one of three variables in it based on the randomness provided by the if-statement, $0 \vee 1 \vee 2$. Here $0 \vee 1 \vee 2$ represents a triple controlled-gate where the control qutrit is the subspaces of the computational basis (visualized in figure 10). The $\log L$ auxiliary qubits are needed for uncomputation.

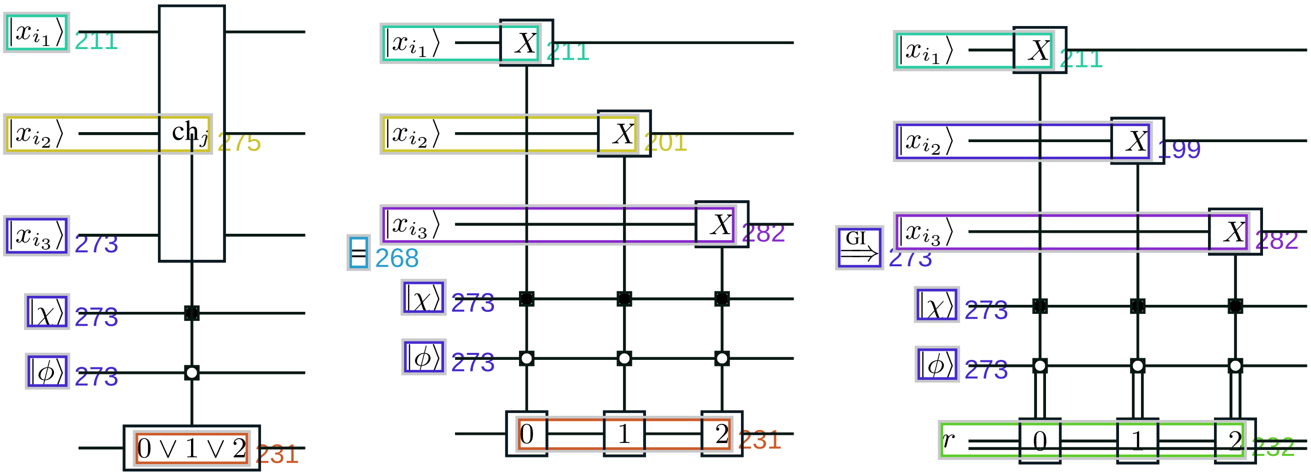

FIG. 10: Implementation of the variable flips of Schöning's walk within amplitude amplification. Here, x_{i_1}, x_{i_2} and x_{i_3} are the variables of the j-th clause. The $0 \vee 1 \vee 2$ block represents a triple controlled-gate where the control qutrit is the subspaces of the basis $|0\rangle$, $|1\rangle$, $|2\rangle$. For the GI algorithm, the walk randomness can be provided by fixing a random number $r \in \{0, 1, 2\}$ for every walk step.

algorithms, why our analysis focuses on the asymptotic run-time rates and coherence-time rates. We argue that these two types of rates are bounded by a general trade-off relation that no hybrid-scheme can beat. For our concrete analysis, we consider hybrid schemes based on Schöning's algorithm, where the latter solves 3-SAT (or more generally k-SAT) problems by random walks in the space of assignments. The walk-procedure allows for several partial Groverization-schemes. We determine the corresponding run-times and coherence-times of these schemes, and demonstrate saturation of the general trade-off relation. Many of these partial Groverizations intuitively lend themselves for efficient circuit implementations, and we provide the main building blocks of these. On a more speculative note, we present numerical evidence that the GI scheme can be partially de-randomized, in the sense that a single 'typical' instance of the classical randomness of the walk appears to mimic the effects of the repeated sampling. This would open for an additional flexibility in the implementation of these hybrid-schemes, still maintaining the optional trade-off.

In this investigation, we have focused on partial Groverizations of Schöning's algorithm. However, this approach should in principle be applicable to any classical probabilistic search scheme, since it essentially only rests on partitions of the underlying randomness. The main concern would be to find 'natural' partitions that are algorithmically accessible, in the sense that the partial Groverization can be implemented efficiently. Explicit run-time and coherence-time rates would also require a classical scheme, as well as partitions, that are sufficiently tractable for analysis, unless one would resort to numerical estimates.

The partial de-randomization of GI-scheme that is suggested by our numerical explorations, would deserve further investigations. In particular, the question is to what extent, and in what sense, the hypothetical relation (30) would be true. Moreover, one may ask if something similar also would apply to fractional GI. For numerical investigations, it would be relevant to extend to larger Hamming distances, further classes of 3-SAT instances, as well as problem sizes. This would likely involve challenges to design reliable numerical estimates, since exact calculations by the very nature of the problem quickly becomes intractable. For purely analytical approaches, some notion of concentration of measure of walks, would be interesting.

In the spirit of [1, 2] we have in this investigation employed 'the walk on $\mathbb{Z}$' as a model of the true Schöning-procedure. In Appendix (see also [6]) we in additionally provide bounds for the true rates of Schöning-procedure and the GW-procedure, in terms of the mirroring processes on $\mathbb{Z}$. It would be relevant to obtain similar bounds also for the GI-process, as well as for the various fractional schemes.

VII. ACKNOWLEDGEMENT

We thank Phillip Keldenich for kindly providing us with the SAT instance that was used for numerical simulations resulted in figure (8). We acknowledge support from Bundesministerium für Bildung und Forschung - BMBF under project QuBRA. The UzK team was also supported by Germany's Excellence Strategy – Cluster of Excellence Matter and Light for Quantum Computing (ML4Q) EXC 2004/1 (390534769).

[1] U. Schöning, *A probabilistic algorithm for k-SAT and constraint satisfaction problems*, in Proceedings of the 40th Annual Symposium on Foundations of Computer Science (IEEE, New York, 1999).

[2] U. Schöning and J. Torán, *The Satisfiability Problem: Algorithms and Analyses* (Lehmanns, Berlin, 2013).

[3] L. K. Grover, *A fast quantum mechanical algorithm for database search*, in Proceedings of the 28th ACM Annual Symposium on the Theory of Computing, STOC 96, p. 212 (ACM, NewYork, 1996).

[4] A. Ambainis, *Quantum search algorithms*, ACM SIGACT News, **35**, 22 (2004).

[5] V. Dunjko, Y. Ge and J. I. Cirac, *Computational speedups using small quantum devices*, PRL **121**, 250501 (2018).

[6] R.A. Moser, *Exact Algorithms for Constraint Satisfaction Problems*, Doctoral thesis, ETH Zürich (2012)

[7] P. Cheeseman, B. Kanefsky and W.M. Taylor, *Where the Really Hard Problems Are.* in Proceedings of the 12th international joint conference on Artificial intelligence - Volume 1, Pages 331–337 (1991)

[8] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd Edition (Wiley-Interscience, New Jersey, 2006)

[9] V. Eshaghian, S. Wilkening, J. Åberg, D. Gross, *Data for explicit run-times for hybrid SAT-solvers*, https://github.com/SoerenWilkening/QuantumSchoening.

[10] H. Callen, *Thermodynamics and an introduction to thermostatistics*, John Wiley, New York, 1985.

[11] Dr. Phillip Keldenich, personal communication, March 2024.

[12] According to [5], Supplemental Material Section B.4, the relative speed-up to the classical Schöning's rate is $f(c) = (1 - \log\sqrt{3})\beta(c)$, where the Beta function up to $O(\frac{\log n}{n})$ is implicitly given as $A\beta(c)\ln\frac{1}{\beta(c)} + B\beta(c) = c$. As mentioned in [5] using a straightforward encoding of each trit into two qubits, one can assume $A = 10$ and $B = 50$. To be consistent with our encoding, we consider $\log_2 3$ qubits to encode a trit and then, calculate the maximum speed-up in the rate, i.e. $f(1) \approx 0.0028$.

Appendix A: From the true Schöning-process to the Markov process on $\mathbb{Z}$

1. The purpose of this appendix

For the calculation of rates, we replace the genuine searches of solutions for 3SAT-problems (the 'true Schöning process'), with a Markovian random walk on the 'Hamming distance' (although we strictly speaking consider a walk on $\mathbb{Z}$). This is analogous to Schönings analysis of the performance of Schönings algorithm [1, 2], where it is argued that this substitute-process yields an upper bound on the rates of the run-time of the algorithm. The purpose of this appendix is to give a more detailed argument for why the success probability of Schönings algorithm is lower bounded by the success-probability of the substitute walk on $\mathbb{Z}$. The reader may also wish to consult [6] for a previous analysis along these lines. Apart from obtaining from bounding the success probability for the true Schöning-process, we also provide the analogous bound for the GW process.

2. The Schöning-process

As described in the main text, the 3SAT problem consists of a collection of clauses $C_1, \ldots, C_L$ on n binary variables, where each clause is of the form $C_j = l_0^{(j)} \vee l_1^{(j)} \vee l_2^{(j)}$, and where each of the literals $l_0^{(j)}, l_1^{(j)}, l_2^{(j)}$ is one of the binary variables, or its negation. The 3SAT formula is the conjunction of all the given clauses, $C := \wedge_{j=1}^{L} C_j$, and the task is to determine whether there exists an assignment $x \in \{0,1\}^{\times n}$ of the n binary variables, which satisfies C. In the following analysis we assume that C either has a *unique* satisfying assignment $x^\star \in \{0,1\}^{\times n}$, or alternatively, that $x^\star$ is *selected* among a set of solutions.

Schönings procedure can be regarded as a stochastic process $(x_l)_{l=0}^{m}$ with $x_l \in \{0,1\}^{\times n}$. The process is initialized by a random assignment x_0 of the bit string, drawn uniformly over all of $\{0,1\}^{\times n}$. On this state it checks all the clauses $C_1, \ldots, C_L$ (according to a pre-determined order). If all are satisfied, then the initial string satisfies C and the algorithm terminates. Otherwise, it finds the first unsatisfied clause, and randomly negates one of the three variables corresponding to the literals of that clause. The algorithm continues according to this random walk until it either finds a satisfying assignment, or it reaches a pre-determined termination-time K. For our purposes, it is convenient to think of the state x_l of the process as a function of a collection of random variables. The initialization is represented by the random variable A, which takes values in $\{0,1\}^{\times n}$. The randomness in the walk is captured by the variables $B = (B_1, \ldots, B_m)$ be random variables where each B_l takes values in $\{0,1,2\}$ (and thus B takes values in $\{0,1,2\}^{\times m}$). Hence, B_l represents one of the three possible choices of which literal to flip at step l. We assume that $A, B_1, \cdots, B_m$ are independent and uniformly distributed, i.e., for $b = (b_1, \ldots, b_m)$ we have

$$\begin{aligned} P(A = a, B = b) =& P(A = a)P(B = b) = P(A = a)P(B_1 = b_1) \cdots P(B_m = b_m), \\ & P(A = a) = \frac{1}{2^n}, \quad \forall a \in \{0,1\}^{\times n} \\ & P(B_l = b_l) = \frac{1}{3}, \quad \forall b_l \in 0,1,2. \end{aligned} \tag{A1}$$

Hence, we can write the Schöning process as $(x_l)_l = (x_l(A,B))_l$, where

$$x_0(a,b) := a, \tag{A2}$$

i.e., a the initial state. At the l:th step of Schöning's process is based on the state x_{l-1} of the previous step. On this state, all the clauses $C_1, \ldots, C_L$ (according to a pre-determined order) are checked. If all are satisfied, then $x_{l-1} = x^\star$ and the process remains in that state, i.e., $x_l = x^\star$. (In other words, t $x^\star$ is an absorbing state for the Schöning-process.) Otherwise, it finds the first unsatisfied clause, which we refer to as C_{j_l}. The selected clause, C_{j_l}, contains the three literals $(l_0^{(j)}, l_1^{(j)}, l_2^{(j)})$. The process constructs x_l by negating the variable corresponding to literal $l_{b_l}^{(j)}$. In other words, it is the l:th component of b that determines which of these three choices that is selected. One may note that the process, by construction, satisfies

$$x_l(a,b) = x_l(a, b_1, \ldots, b_l). \tag{A3}$$

Hence, the value of $x_l(a,b)$ only depends on the values of $b_1, \ldots, b_l$, not any of the 'later' variables $b_{l+1}, b_{l+1}, \ldots$. One may also note that $A, B_1, B_2, \ldots, B_K$ encompasses *all* the randomness in the process. In other words, the state x_l is uniquely determined by $a, b_1, \ldots, b_l$.

3. The proof idea

As described above, the true Schöning-process $(x_l)_l$ is a walk on bit-strings. However, for the analysis of the optimal rates, we follow in the steps of Schöning [1, 2], and instead focus on the Hamming-distance to the (selected) solution $x^\star$. In principle, nothing prevents us from projecting the state x_l of the Schöning-process, to the Hamming distance $d_H(x_l,x^\star)$ (i.e. projecting onto $\mathbb{N}$). However, this would generally yield a process that would be no easier to analyze than the original Schöning-process. One may for example note that although Schönings-process $(x_l)_l$ is Markovian on the space of bit-strings, one cannot generally expect its projection $\big(d_H(x_l,x^\star)\big)_l$ to be Markovian on $\mathbb{N}$. The general idea for the analysis is to replace (via a coupling) the true projection $\big(d_H(x_l,x^\star)\big)_l$ with another process $(\tilde{d}_l)_l$ on $\mathbb{N}$, which is Markovian and which moreover upper-bounds the true Hamming-distance, $d_H(x_l,x^\star)\leq\tilde{d}_l$. One may note that the Schöning-process is 'successful' if it finds the solution $x^\star$. Hence, we can express the success probability at step l as $P(x_l=x^\star)=P\big(d_H(x_l,x^\star)=0\big)$. From the bound $d_H(x_l,x^\star)\leq\tilde{d}_l$ it follows that $P(x_l=x^\star)\geq P(\tilde{d}_l=0)$. In other words, the success-probability of the Schöning-process is lower-bounded by the probability that the substitute-process $\tilde{d}_l$ reaches 0. The fact that $(\tilde{d}_l)_l$ is Markovian makes the analysis more tractable. However, the value 0 corresponds to an absorbing boundary. (If we find the solution at an earlier stage, we should terminate the process rather than walking on.) To further ease the analysis, we remove this boundary and instead introduce yet another walk $(d_l)_l$ on $\mathbb{Z}$, which we regard as 'successful' whenever $d_l\leq 0$. For this process we moreover establish the bound $P(\tilde{d}_l=0)\geq P(d_l\leq 0)$, and thus $P(x_l=x^\star)\geq P(d_l\leq 0)$. By the trivial bound $P(d_l\leq 0)\geq P(d_l=0)$, we thus ultimately get the bound $P(x_l=x^\star)\geq P(d_l=0)$. For the calculation of the optimal rates, our starting point is an expression for $P(d_l=0)$. By the inequality $P(x_l=x^\star)\geq P(d_l=0)$ it follows that the calculated rates are upper bounds to the true rates of the Schöning-process.

4. Constructing a walk $(\tilde{d}_l)_l$ on $\mathbb{N}$ such that $d_H(x_l,x^\star)\leq\tilde{d}_l$

Related to the Schöning-process $(x_l)_l$, we here wish to construct another process $(\tilde{d}_l)_l$, where $\tilde{d}_l$ takes values in $\mathbb{N}$ for all $l\in\mathbb{N}$, and is such that

$$d_H\big(x_l(a,b_1,\ldots,b_l),x^\star\big)\leq\tilde{d}_l(a,b_1,\ldots,b_l),\quad\forall a\in\{0,1\},\quad\forall b\in\{0,1,2\}^{\times m},\quad l=0,1,2,\ldots,m. \tag{A4}$$

In other words, we want to make sure that $\tilde{d}_l$ *always* is an upper bound to the Hamming distance between x_l and $x^\star$. This requires a considerable coordination between the two processes. In particular, whenever x_l moves in the 'wrong' direction (i.e. increases the Hamming distance to $x^\star$) then d_l also has to increase. To this end, we consider the list of clauses $C_1,\ldots,C_L$. For each clause C_j it is the case that $C_j(x^\star)=1$. Hence, for each j, at least one of the literals $l_0^{(j)}, l_1^{(j)}, l_2^{(j)}$ is satisfied by $x^\star$. Among these satisfied clauses we select one of these satisfied literals, and let $r_j\in\{0,1,2\}$ be its index. In other words, we are guaranteed that $l_j^{(r_j)}(x^\star)=1$.

As already described above, the Schöning-process $(x_l(a,b))_l$ is uniquely determined by $(a,b_1,\ldots,b_l)$, and does in turn uniquely determines the unsatisfied clauses C_{j_l}, as long as $x_l(a,b_1,\ldots,b_l)\neq x^\star$. Consequently, it also uniquely determines a sequence of 'selected' literals r_{j_l}, whenever $x_l(a,b_1,\ldots,b_l)\neq x^\star$. For each $l\in 1,2,\ldots$, we define a mapping $(a,b_1,\ldots b_{l-1})\mapsto f_l(a,b_1,\ldots,b_{l-1})\in\{0,1,2\}$ by

$$\begin{aligned}
f_1(a)&:=\begin{cases}0 & \text{if } x_0\equiv a=x^\star,\\ r_{j_1} & \text{if } x_0\equiv a\neq x^\star.\end{cases}\\
f_l(a,b_1,\ldots,b_{l-1})&:=\begin{cases}0 & \text{if } x_{l-1}(a,b_1,\ldots,b_{l-1})=x^\star,\\ r_{j_l} & \text{if } x_{l-1}(a,b_1,\ldots,b_{l-1})\neq x^\star.\end{cases}\qquad l=2,3,\ldots
\end{aligned}\tag{A5}$$

The purpose of $f_l(a,b_1,\ldots,b_{l-1})$ is to determine which value of b_l that should correspond to a 'successful' move for the $(\tilde{d}_l)_l$-process. More precisely, we define $(\tilde{d}_l(a,b))_l$ by

$$\begin{aligned}
\tilde{d}_0(a,b)&:=d_H\big(x_0(a,b),x^\star\big)=d_H(a,x^\star),\\
\tilde{d}_l(a,b_1,\ldots,b_l)&:=\begin{cases}0 & \text{if}\quad \tilde{d}_{l-1}(a,b_1,\ldots,b_{l-1})=0\\ \tilde{d}_{l-1}(a,b_1,\ldots,b_{l-1})+1 & \text{if}\quad \tilde{d}_{l-1}(a,b_1,\ldots,b_{l-1})\neq 0,\quad b_l\neq f_l(a,b_1,\ldots,b_{l-1})\\ \tilde{d}_{l-1}(a,b_1,\ldots,b_{l-1})-1 & \text{if}\quad \tilde{d}_{l-1}(a,b_1,\ldots,b_{l-1})\neq 0,\quad b_l=f_l(a,b_1,\ldots,b_{l-1})\end{cases}\quad l=1,2,\ldots
\end{aligned}\tag{A6}$$

In words, the first condition in the bracket means that 0 is an absorbing state, i.e., if $\tilde{d}_l(a,b)=0$ for some l, then $\tilde{d}_{l'}(a,b)=0$ for all $l'>l$. The other two cases make sure that the $\tilde{d}_l$ moves in 'coordination' with the Schöning-process $(x_l(a,b))_l$, in such a manner that it is guaranteed that $d_H\big(x_l(a,b_1,\ldots,b_l),x^\star\big)$ does not increase above $\tilde{d}_l(a,b_1,\ldots,b_l)$.

Lemma 1. *The Schöning process $(x_l)_{l\in\mathbb{N}}$ and the process $(\tilde{d}_l)_{l\in\mathbb{N}}$ as defined by (A5) and (A6) satisfy*

$$d_H\big(x_l(a,b_1,\ldots,b_l),x^\star\big)\leq\tilde{d}_l(a,b_1,\ldots,b_l),\quad\forall a\in\{0,1\}^{\times n},\quad\forall b\in\{0,1,2\}^{\times l},\quad l=0,1,2,\ldots.\tag{A7}$$

One may note that (A7) holds for every single element in the event-space, and Lemma 1 does thus not depend on the actual probability distribution of $A, B_1, \dots, B_l$. However, there are other steps in our proofs that do depend crucially on these variables being independent and uniformly distributed.

Proof. We first note that

$$x_0(a,b) = a, \quad \tilde{d}_0 = d_H(a, x^\star) \tag{A8}$$

and thus (A7) is satisfied for $l = 0$ for all a, b.

Now, assume that (A7) holds for some $l-1, a, b$. We have the following cases:

- **Case** $x_{l-1}(a,b) = x^\star$**:** Since we assume that $x^\star$ is absorbing, it follows that $x_l(a,b) = x^\star$ and consequently $d_H\big(x_l(a,b_1,\dots,b_l), x^\star\big) = 0$. Concerning d_{l-1}, we can distinguish yet two sub-cases:
  - **Case** $\tilde{d}_{l-1}(a,b_1,\dots,b_{l-1}) = 0$**:** By construction (first case in (A6)) $\tilde{d}_l(a,b_1,\dots,b_l) = 0$, and (A7) is thus satisfied for l, a, b.
  - **Case** $\tilde{d}_{l-1}(a,b_1,\dots,b_{l-1}) \neq 0$**:** Then $\tilde{d}_{l-1}(a,b_1,\dots,b_{l-1}) \geq 1$. Since the process d can change at most one step, it follows that $d_{l-1}(a,b_1,\dots,b_{l-1}) \geq 0$, and thus (A7) is satisfied for l, a, b.
- **Case** $x_{l-1}(a,b) \neq x^\star$**:** Since we assume that (A7) holds for $l-1, a, b$ it follows that $\tilde{d}_{l-1}(a,b_1,\dots,b_{l-1}) \geq 1$. Moreover, since $x_{l-1}(a,b) \neq x^\star$, we have $f_l(a,b_1,\dots,b_{l-1}) = r_{j_l}$. Again, we can distinguish two sub-cases:
  - **Case** $f_l(a,b_1,\dots,b_{l-1}) = b_l$**:** In this case, the d-process decreases one step. However, by construction r_{j_l} is one of the 'successful' flips for the Schöning-process, hence the x-process also decreases one step. By assumption the inequality (A7) is satisfied for $l-1, a, b$, and since both the x-process and the d-process decrease one step, (A7) remains satisfied for l, a, b.
  - **Case** $f_l(a,b_1,\dots,b_{l-1}) \neq b_l$**:** In this case, the d-process increases one step. The x-process may increase or decrease, but with at most one step, so (A7) remains satisfied for l, a, b.

By induction, we can conclude that (A7) is satisfied for all l, a, b. □

5. $(\tilde{d}_l)_l$ is a Markov chain

In the following we wish to show that $(\tilde{d}_l)_l$ is a Markov chain. Recall that both the genuine Schöning-process $(x_l)_l$, as well as the walk $(\tilde{d}_l)_l$, are determined by a sequence of 'walk variables' $(B_l)_l$ (and initial-state variable A). The Schöning-walk itself is Markovian, but it is *a priori* not obvious that the process $(\tilde{d}_l)_l$ also is Markovian, particularly since the l-th step of the latter is determined by a complicated function of all the walk-variables up to the l-th step, as described by (A6). However, in spite appearances, it turns out that (A6) defines a mapping from the original set of random variables $(B_l)_l$ to a new set of variables $(\tilde{B}_l)_l$, in such a manner that the *change* from $\tilde{d}_{m-1}$ to $\tilde{d}_m$ is determined by $\tilde{B}_m$, and *only* by $\tilde{B}_m$. Moreover, it turns out that $(\tilde{B}_l)_l$ is an iid sequence. Since all the $\tilde{B}_l$ are independent, it follows that $(\tilde{d}_l)_l$ must be a Markov chain. In order to show that the new sequence of variables $(\tilde{B}_l)_l$ is iid, what we actually do is to show that (A6) induces a bijection on $\{0,1\}^{\times n} \times \{0,1,2\}^l$. Since $\big(A, (B_l)_l\big)$ is uniformly distributed (see (A1)) it follows by the bijection that $\big(A, (\tilde{B}_l)_l\big)$ also is uniformly distributed, and thus in particular that $(\tilde{B}_l)_l$ is iid.

6. A bijection

The following lemma introduces functions f_s. Later, in the proof of Proposition 6, we will let these mappings be the functions $f_l(a,b_1,\dots,b_{l-1})$ in (A5). Since the latter are algorithmically defined, via the Schöning-process $(x_l)_l$, it is challenging to get a hold on the properties of these mappings. It is thus worth noting that (apart from domains and ranges) Lemma 3 (and Lemma 5) makes no assumptions on the properties of the mappings f_s. Hence, our lack of control over the mappings $f_l(a,b_1,\dots,b_{l-1})$ will not be an issue in the subsequent proofs.

As preparation, we make the following observations.

Lemma 2. *If* $t, t', r \in \{0,1,2\}$*, then*

$$(t - r) \bmod 3 = (t' - r) \bmod 3 \quad \Leftrightarrow \quad t = t'. \tag{A9}$$

Moreover, if $t, r \in \{0,1,2\}$, then

$$\big((t+r) \bmod 3 - r\big) \bmod 3 = t. \tag{A10}$$

Lemma 3. *Let $f_1 : \{0,1\}^{\times n} \to \{0,1,2\}$ and $f_s : \{0,1\}^{\times n} \times \{0,1,2\}^{\times (s-1)} \to \{0,1,2\}$ for $s = 2, \dots, l$ be given. Define the mapping $\{0,1\}^{\times n} \times \{0,1,2\}^{\times l} \ni (b_1, \dots, b_l) \mapsto Q(a, b_1, \dots, b_l) = (\tilde{a}, \tilde{b}_1, \dots, \tilde{b}_l) \in \{0,1\}^{\times n} \times \{0,1,2\}^{\times l}$ by*

$$\begin{aligned} \tilde{a} &:= a, \\ \tilde{b}_1 &:= \big(b_1 - f_1(a)\big) \bmod 3, \\ \tilde{b}_2 &:= \big(b_2 - f_2(a, b_1)\big) \bmod 3, \\ \tilde{b}_3 &:= \big(b_3 - f_3(a, b_1, b_2)\big) \bmod 3, \\ &\vdots \\ \tilde{b}_l &:= \big(b_l - f_l(a, b_1, \dots, b_{l-1})\big) \bmod 3. \end{aligned} \tag{A11}$$

Then Q is a bijection on $\{0,1\}^{\times n} \times \{0,1,2\}^{\times l}$.

Proof. To show that Q is a bijection, we first show that it is injective, and then that it is surjective.

Let $(a, b_1, \dots, b_l), (a', b'_1, \dots, b'_l) \in \{0,1\}^{\times n} \times \{0,1,2\}^{\times l}$ be such that

$$Q(a, b_1, \dots, b_l) = Q(a', b'_1, \dots, b'_l). \tag{A12}$$

By the first line of (A11) it follows that

$$a = \tilde{a} = a'. \tag{A13}$$

By the second line of (A11) it follows that

$$\big(b_1 - f_1(a)\big) \bmod 3 = \big(b'_1 - f_1(a')\big) \bmod 3, \tag{A14}$$

which combined with (A13) yields

$$\big(b_1 - f_1(a)\big) \bmod 3 = \big(b'_1 - f_1(a)\big) \bmod 3. \tag{A15}$$

Since $f_1(a), b_1, b'_1 \in \{0,1,2\}$ it follows by (A9) that

$$b_1 = b'_1. \tag{A16}$$

As an induction hypothesis, assume that for some $s \geq 2$, it is the case that

$$a = a', \quad b_j = b'_j, \quad j = 1, \dots, s-1. \tag{A17}$$

The sth line of (A12) implies

$$\big(b_s - f_s(a, b_1, \dots, b_{s-1})\big) \bmod 3 = \big(b'_s - f_s(a', b'_1, \dots, b'_{s-1})\big) \bmod 3. \tag{A18}$$

By the induction hypothesis, this implies

$$\big(b_s - f_s(a, b_1, \dots, b_{s-1})\big) \bmod 3 = \big(b'_s - f_s(a, b_1, \dots, b_{s-1})\big) \bmod 3. \tag{A19}$$

Since $b_s, b'_s, f_s(a, b_1, \dots, b_{s-1}) \in \{0,1,2\}$, it follows by (A9) that

$$b_s = b'_s. \tag{A20}$$

Since the induction hypothesis is true for $s = 2$, we can conclude that it is true for all $s = 2, \dots, l$. We can thus conclude that Q is injective.

Next we wish to show that Q is surjective onto $\{0,1\}^{\times n} \times \{0,1,2\}^{\times l}$. Let $(\tilde{a}', \tilde{b}'_1, \dots, \tilde{b}'_l) \in \{0,1\}^{\times n} \times \{0,1,2\}^{\times l}$. Define

$$\begin{aligned} a &:= \tilde{a}', \\ b_1 &:= \big(\tilde{b}'_1 + f_1(\tilde{a}')\big) \bmod 3 = \big(\tilde{b}'_1 + f_1(\tilde{a}')\big) \bmod 3, \end{aligned} \tag{A21}$$

and the sequence $(b_j)_{j=2}^{l}$ recursively by

$$b_j := \big(\tilde b'_j + f_j(a, b_{j-1}, \dots, b_1)\big) \bmod 3, \quad j = 2, \dots, l, \tag{A22}$$

for a' and b'_1 as defined in (A21). In the following, we wish to show that $Q(a, b_1, \dots, b_l) = (\tilde a', \tilde b'_1, \dots, \tilde b'_l)$. For notational convenience, we introduce the components $Q_0(a, b_1, \dots, b_l) := \tilde a$ and $Q_j(a, b_1, \dots, b_l) := \tilde b_j$, quad $j = m2, \dots, l$.

By the first line of (A11) we have

$$Q_0(a, b_1, \dots, b_l) = a = \tilde a'. \tag{A23}$$

By the second line of (A11)

$$\begin{aligned} Q_1(a, b_1, \dots, b_l) &= \big(b_1 - f_1(a)\big) \bmod 3 \\ &= \Big(\big(\tilde b'_1 + f_1(\tilde a')\big) \bmod 3 - f_1(a)\Big) \bmod 3 \\ &= \Big(\big(\tilde b'_1 + f_1(\tilde a')\big) \bmod 3 - f_1(\tilde a')\Big) \bmod 3 \\ &\quad \text{[By (A10)]} \\ &= \tilde b'_1. \end{aligned} \tag{A24}$$

For all $j \geq 2$ we moreover have

$$\begin{aligned} Q_j(a, b_1, \dots, b_l) &= \big(b_j - f_j(a, b_1, \dots, b_{j-1})\big) \bmod 3 \\ &\quad \text{[By (A22)]} \\ &= \Big(\big(\tilde b'_j + f_j(a, b_{j-1}, \dots, b_1)\big) - f_j(a, b_1, \dots, b_{j-1})\Big) \bmod 3 \\ &\quad \text{[By (A10)]} \\ &= \tilde b'_j. \end{aligned} \tag{A25}$$

Hence, we can conclude that $Q(a, b_1, \dots, b_l) = (\tilde a', \tilde b'_1, \dots, \tilde b'_l)$. Hence Q is surjective, and thus bijective. □

7. Transformations that preserve uniformity

We make the following basic observation

Lemma 4. *Let $\mathcal{S}$ be some finite set. Let $Q : \mathcal{S} \to \mathcal{S}$ be invertible. Let R be some random variable on $\mathcal{S}$. If R is uniformly distributed over $\mathcal{S}$, the $Q(R)$ is also uniformly distributed over $\mathcal{S}$.*

Proof.

$$P(Q(R) = s) = P(R = Q^{-1}(s)) = \frac{1}{|\mathcal{S}|} \tag{A26}$$

□

Lemma 5. *Let $f_1 : \{0,1\}^{\times n} \to \{0,1,2\}$ and $f_s : \{0,1\}^{\times n} \times \{0,1,2\}^{\times (s-1)} \to \{0,1,2\}$ for $s = 2, \dots, l$ be given. Assume that $B_1, \dots, B_l$ are random variables that take values in $\{0,1,2\}$, A be a random variable that takes values in $\{0,1\}^{\times n}$, and that these are distributed as*

$$P(A = a, B_1 = b_1, \dots, B_l = b_l) = \frac{1}{2^n 3^l}, \quad \forall a \in \{0,1\}^{\times n}, \quad \forall (b_1, \dots, b_l) \in \{0,1,2\}^{\times l}. \tag{A27}$$

Define $\tilde B_1, \dots, \tilde B_l$ by

$$\begin{aligned} \tilde B_1 &:= \big(B_1 - f_1(A)\big) \bmod 3, \\ \tilde B_2 &:= \big(B_2 - f_2(A, B_1)\big) \bmod 3, \\ \tilde B_3 &:= \big(B_3 - f_3(A, B_1, B_2)\big) \bmod 3, \\ &\vdots \\ \tilde B_l &:= \big(B_l - f_l(A, B_1, \dots, B_{l-1})\big) \bmod 3. \end{aligned} \tag{A28}$$

Then

$$P(A=a,\tilde B_1=\tilde b_1,\dots,\tilde B_l=\tilde b_l)=\frac{1}{2^n3^l},\quad \forall a\in\{0,1\}^{\times n},\quad \forall(\tilde b_1,\dots,\tilde b_l)\in\{0,1,2\}^{\times l}. \tag{A29}$$

Consequently, $A,\tilde B_1,\dots,\tilde B_l$ are independent and uniformly distributed.

Proof. By (A27) we know that $(A,B_1,\dots,B_l)$ is uniformly distributed on $\{0,1\}^{\times n}\times\{0,1,2\}^{\times l}$. With the additional definition $\tilde A:=A$, we note that (A28) can be rewritten as

$$(\tilde A,\tilde B_1,\dots,\tilde B_l):=Q(A,B_1,\dots,B_l), \tag{A30}$$

where $Q:\{0,1\}^{\times n}\times\{0,1,2\}^{\times l}\to\{0,1\}^{\times n}\times\{0,1,2\}^{\times l}$ is as defined in Lemma 3. By Lemma 3, we moreover know that Q is a bijection on $\{0,1\}^{\times n}\times\{0,1,2\}^{\times l}$ and thus invertible. Hence, by Lemma A29, we know that $(\tilde A,\tilde B_1,\dots,\tilde B_l)$ also is uniformly distributed over $\{0,1\}^{\times n}\times\{0,1,2\}^{\times l}$. Since $\tilde A=A$, we can conclude that (A29) holds.

By (A29) i follows that

$$\begin{aligned}&P(A=a,\tilde B_1=\tilde b_1,\dots,\tilde B_l=\tilde b_l)=P(A=a)P(\tilde B_1=\tilde b_1)\cdots P(\tilde B_l=\tilde b_l),\\&P(A=a)=\frac{1}{2^n},\quad P(\tilde B_1=\tilde b_1)=\frac{1}{3},\dots,P(\tilde B_l=\tilde b_l)=\frac{1}{3}.\end{aligned} \tag{A31}$$

and thus $A,\tilde B_1,\dots,\tilde B_l$ are independent and uniformly distributed.

□

8. The process $(\tilde d_l)_l$ is a Markov chain

Proposition 6. *Let $(\tilde d_l)_l$ be the process as defined by (A5) and (A6), with respect to the variables $A, B_1, B_2,\dots$ distributed as in (A1). For each m there exist variables $\tilde B_1,\cdots,\tilde B_m$ that are iid and uniformly distributed on $\{0,1,2\}$, and are independent of A, such that*

$$\begin{aligned}\tilde d_0:=&d_H(A,x^\star),\\ \tilde d_l:=&\begin{cases}0 & \text{if}\quad \tilde d_{l-1}=0\\ \tilde d_{l-1}+1 & \text{if}\quad \tilde d_{l-1}\neq 0,\quad \tilde B_l\neq 0\\ \tilde d_{l-1}-1 & \text{if}\quad \tilde d_{l-1}\neq 0,\quad \tilde B_l=0\end{cases}\quad l=1,2,\dots,m\end{aligned} \tag{A32}$$

Hence, $(\tilde d_l)_l$ is a Markov chain described by the transition probabilities

$$P(\tilde d_{l+1}=j|\tilde d_l=k)=\delta_{j,0}\delta_{k,0}+(1-\delta_{k,0})(\frac{1}{3}\delta_{j,k-1}+\frac{2}{3}\delta_{j,k+1}),\quad \forall j,k\in\mathbb{N},\quad \forall l \tag{A33}$$

with initial distribution

$$P(\tilde d_0=j)=P\big(d_H(A,x^\star)=j\big). \tag{A34}$$

Moreover, for the distribution of A as in (A1) we have

$$P(\tilde d_0=j)=\begin{cases}\frac{1}{2^n}\binom{n}{j}, & 0\leq j\leq n\\ 0 & \text{otherwise}\end{cases} \tag{A35}$$

In (A33), the term $\delta_{j,0}\delta_{k,0}$ signifies $d=0$ being an absorbing state. In the second term, the effect of the factor $(1-\delta_{k,0})$ is that if the chain is not in the absorbing state, then the transition probabilities are given by $\frac{1}{3}\delta_{j,k-1}+\frac{2}{3}\delta_{j,k+1}$. Hence, with probability $1/3$, it takes a step 'down', and with probability $2/3$ it takes a step 'up'.

Proof. For $t,r\in\{0,1,2\}$ it is the case that

$$t=r\quad\Leftrightarrow\quad (t-r)\bmod 3=0. \tag{A36}$$

By this observation, it follows that (A6) can be rewritten as

$$\begin{aligned} \tilde{d}_0(a,b) &:= d_H(a,x^\star), \\ \tilde{d}_l(a,b_1,\ldots,b_l) &:= \begin{cases} 0 & \text{if} \quad \tilde{d}_{l-1}(a,b_1,\ldots,b_{l-1}) = 0 \\ \tilde{d}_{l-1}(a,b_1,\ldots,b_{l-1}) + 1 & \text{if} \quad \tilde{d}_{l-1}(a,b_1,\ldots,b_{l-1}) \neq 0, \quad \big(b_l - f_l(a,b_1,\ldots,b_{l-1})\big) \bmod 3 \neq 0 \\ \tilde{d}_{l-1}(a,b_1,\ldots,b_{l-1}) - 1 & \text{if} \quad \tilde{d}_{l-1}(a,b_1,\ldots,b_{l-1}) \neq 0, \quad \big(b_l - f_l(a,b_1,\ldots,b_{l-1})\big) \bmod 3 = 0 \end{cases} \quad l = 1,2,\ldots \end{aligned} \tag{A37}$$

Next, we rewrite (A37) such that we suppress the explicit dependence on the elementary events (a,b).

$$\begin{aligned} \tilde{d}_0 &:= d_H(A,x^\star), \\ \tilde{d}_l &:= \begin{cases} 0 & \text{if} \quad \tilde{d}_{l-1} = 0 \\ \tilde{d}_{l-1} + 1 & \text{if} \quad \tilde{d}_{l-1} \neq 0, \quad \big(B_l - f_l(A,B_1,\ldots,B_{l-1})\big) \bmod 3 \neq 0 \\ \tilde{d}_{l-1} - 1 & \text{if} \quad \tilde{d}_{l-1} \neq 0, \quad \big(B_l - f_l(A,B_1,\ldots,B_{l-1})\big) \bmod 3 = 0 \end{cases} \quad l = 1,2,\ldots \end{aligned} \tag{A38}$$

If we define $\tilde{B}_1,\ldots,\tilde{B}_l$ by

$$\begin{aligned} \tilde{B}_1 &:= f_1(A), \\ \tilde{B}_2 &:= \big(B_2 - f_2(A,B_1)\big) \bmod 3, \\ \tilde{B}_3 &:= \big(B_3 - f_3(A,B_1,B_2)\big) \bmod 3, \\ &\vdots \\ \tilde{B}_l &:= \big(B_l - f_l(A,B_1,\ldots,B_{l-1})\big) \bmod 3, \end{aligned} \tag{A39}$$

then we can rewrite (A38) as

$$\begin{aligned} \tilde{d}_0 &:= d_H(A,x^\star), \\ \tilde{d}_l &:= \begin{cases} 0 & \text{if} \quad \tilde{d}_{l-1} = 0 \\ \tilde{d}_{l-1} + 1 & \text{if} \quad \tilde{d}_{l-1} \neq 0, \quad \tilde{B}_l \neq 0 \\ \tilde{d}_{l-1} - 1 & \text{if} \quad \tilde{d}_{l-1} \neq 0, \quad \tilde{B}_l = 0 \end{cases} \quad l = 1,2,\ldots \end{aligned} \tag{A40}$$

By Lemma 5 we know that $A, \tilde{B}_1,\ldots,\tilde{B}_l$ are independent and uniformly distributed. Since the l:th step is determined solely by $\tilde{B}_l$, and these are independent of each other, and of A, it follows that $(d_l)_l$ is a Markov chain. By inspecting (A40) we first see that

$$P(\tilde{d}_l = j | \tilde{d}_{l-1} = 0) = \delta_{j,0}, \tag{A41}$$

while for $\tilde{d}_{l-1} = k \neq 0$ we have

$$\begin{aligned} P(\tilde{d}_l = j | \tilde{d}_{l-1} = k) &= \delta_{j,k+1} P(\tilde{B}_l \neq 0) + \delta_{j,k-1} P(\tilde{B}_l = 0) \\ &= \frac{2}{3}\delta_{j,k+1} + \frac{1}{3}\delta_{j,k-1}, \end{aligned} \tag{A42}$$

where the last step follows since each $\tilde{B}_l$ is uniformly distributed over $\{0,1,2\}$. By combining the cases (A41) and (A42) we obtain (A33). By (A40) it moreover follows that $P(\tilde{d}_0 = j) = P\big(d_H(A,x^\star) = j\big)$. Since A is uniformly distributed over $\{0,1\}^{\times n}$, it means that $d_H(A,x^\star)$ is binomially distributed. Thus for $0 \leq j \leq n$, we have $P(\tilde{d}_0 = j) = \frac{1}{2^n}\binom{n}{j}$. □

9. Relating probabilities of $(x_l)_l$ and $(\tilde{d}_l)_l$

The reason for why we introduce the walk $(\tilde{d}_l)_l$ is in order to bound the relevant success-probabilities of the more complicated true Schöning-walk $(x_l)_l$. The lemma below considers two such inequalities, which we will use when we determine the bounds for the Groverized walk.

Lemma 7. *Let $(x_l)_{l\in\mathbb{N}}$ be the Schöning process for bit strings of length n, with $x^\star$ the selected satisfying assignment. Let $(\tilde{d}_l)_l$ be the process as defined by (A5) and (A6). Then*

$$P(x_m = x^\star) \geq P(\tilde{d}_m = 0), \tag{A43}$$

$$P(x_m = x^\star | d_H(x_0, x^\star) = j) \geq P(\tilde{d}_m = 0 | \tilde{d}_0 = j), \tag{A44}$$

Proof. We begin by proving inequality (A43). For the sake of notational simplicity, we let ω denote the elements of the event space (where we could regard ω as (a, b) or $(a, \tilde{b})$). By Lemma 1 we know that

$$d_H(x_m(\omega), x^\star) \leq \tilde{d}_m(\omega), \tag{A45}$$

which implies

$$\tilde{d}_m(\omega) = 0 \quad \Rightarrow \quad d_H(x_m(\omega), x^\star) = 0 \tag{A46}$$

and thus

$$\{\omega : \tilde{d}_m(\omega) = 0\} \subset \{\omega : d_H(x_m(\omega), x^\star) = 0\} = \{\omega : x_m(\omega) = x^\star\} \tag{A47}$$

and thus

$$\begin{aligned} P(\tilde{d}_m = 0) &= P(\{\omega : \tilde{d}_m(\omega) = 0\}) \\ &\leq P(\{\omega : x_m(\omega) = x^\star\}) \\ &= P(x_m = x^\star), \end{aligned} \tag{A48}$$

which proves (A43).

We next turn to the proof of (A44) By definition of the walk $(\tilde{d}_l)_l$ we have $\tilde{d}_0(\omega) = d_H(x_0(\omega), x^\star)$, and thus

$$\{\omega : \tilde{d}_0(\omega) = j\} = \{\omega : d_H(x_0(\omega), x^\star) = j\} \tag{A49}$$

and consequently

$$P(\tilde{d}_0 = j) = P(d_H(x_0, x^\star) = j). \tag{A50}$$

By combining (A47) and (A49) we obtain

$$\{\omega : \tilde{d}_m(\omega) = 0\} \cap \{\omega : \tilde{d}_0(\omega) = j\} \subset \{\omega : x_m(\omega) = x^\star\} \cap \{\omega : d_H(x_0(\omega), x^\star) = j\} \tag{A51}$$

and consequently

$$\begin{aligned} P(\tilde{d}_m = 0, \tilde{d}_0 = j) =& P(\{\omega : \tilde{d}_m(\omega) = 0\} \cap \{\omega : \tilde{d}_0(\omega) = j\}) \\ \leq& P(\{\omega : x_m(\omega) = x^\star\} \cap \{\omega : d_H(x_0(\omega), x^\star) = j\}) \\ =& P(x_m = x^\star, d_H(x_0, x^\star) = j). \end{aligned} \tag{A52}$$

By combining this with (A50) we can conclude that

$$P(\tilde{d}_m = 0 | \tilde{d}_0 = j) \leq P(x_m = x^\star | d_H(x_0, x^\star) = j), \tag{A53}$$

which proves (A44). □

10. From walks on $\mathbb{N}$ to walks on $\mathbb{Z}$

So far, we have replaced the projection of the Schöning-process $(x_l)_l$ to the Hamming distance $d_H(x_m, x^\star)$ with the substitute Markov-chain $(\tilde{d}_l)_l$. Similar to $x^\star$ being an absorbing state of (x_l), the process $(\tilde{d}_l)_l$ has 0 as absorbing state. As a model of the true Schöning process, this absorbing state certainly makes sense, since it corresponds to a setting where we at each step monitor whether a solution has been reached, and the process is terminated once this happens. For the sake of obtaining tractable expressions for the relevant probabilities, we here take one step further and instead consider a walk on $\mathbb{Z}$. Analogously to how Lemma 7 bounds the relevant probabilities of the true Schöning process, with the corresponding quantities in $(d_l)_l$, Lemma 8 below, bounds the relevant probabilities of $(\tilde{d}_l)_l$ in terms of corresponding quantities for a Markov-chain $(d_l)_l$ extended to the whole of $\mathbb{Z}$.

As a bit of a side remark, one may note that the results in (8) does not necessarily refer to the particular Markov-chain defined by (A5) and (A6), but could be any Markov chain on $\mathbb{N}$ with fixed transition probabilities and absorbing boundary condition at 0.

Lemma 8. *Let $(\tilde{d}_l)_{l\in\mathbb{N}}$ be a Markov chain on $\mathbb{N}$, with transition probabilities*

$$P(\tilde{d}_{l+1} = j|\tilde{d}_l = k) = \delta_{j,0}\delta_{k,0} + (1-\delta_{k,0})\big((1-q)\delta_{j,k-1} + q\delta_{j,k+1}\big), \quad \forall j,k \in \mathbb{N}, \quad \forall l \in \mathbb{N}, \tag{A54}$$

for some $0 \leq q \leq 1$. Let $(d_l)_{l\in\mathbb{N}}$ be a Markov chain on $\mathbb{Z}$, with transition probabilities

$$P(d_{l+1} = j|d_l = k) = (1-q)\delta_{j,k-1} + q\delta_{j,k+1}, \quad \forall j,k \in \mathbb{Z}, \quad \forall.l \in \mathbb{N}. \tag{A55}$$

Then

$$P\big(d_m \leq 0|d_0 = j\big) \leq P\big(\tilde{d}_m = 0|\tilde{d}_0 = j\big), \quad \forall m \in \mathbb{N}, \quad \forall j \in \mathbb{N}. \tag{A56}$$

Consequently, if the initial state d_0 is such that

$$P(d_0 = j) = \begin{cases} P(\tilde{d}_0 = j), & j \geq 0 \\ 0, & j < 0 \end{cases} \tag{A57}$$

then

$$P(d_m = 0) \leq P(d_m \leq 0) \leq P(\tilde{d}_m = 0), \quad \forall m \in \mathbb{N}. \tag{A58}$$

Proof. For notational convenience, we define

$$M_{j,k} := P(\tilde{d}_{l+1} = j|\tilde{d}_l = k), \tag{A59}$$

and

$$\tilde{M}_{j,k} := P(d_{l+1} = j|d_l = k). \tag{A60}$$

By comparing with (A54) and (A55) one can see that

$$M_{j,k} = \tilde{M}_{j,k}, \quad \forall j > 0, \quad \forall k > 0. \tag{A61}$$

We note that 0 is an absorbing state for $(\tilde{d}_l)_l$. Hence,

$$\tilde{d}_{s-1} = 0 \quad \Rightarrow \quad \tilde{d}_s = 0, \tag{A62}$$

which implies

$$\tilde{d}_s > 0 \quad \Rightarrow \quad \tilde{d}_{s-1} > 0, \tag{A63}$$

which in turn implies

$$P(\tilde{d}_s = k_s|\tilde{d}_{s-1} = 0) = 0, \quad \text{if} \quad k_s > 0. \tag{A64}$$

We begin by proving (A56). For this purpose, assume that $j > 0$.

$$\begin{aligned}
& P(\tilde{d}_l > 0|\tilde{d}_0 = j) \\
&= \sum_{k_l>0} P(\tilde{d}_l = k_l|\tilde{d}_0 = j) \\
& [\text{By Markovianity}] \\
&= \sum_{k_l>0} \sum_{k_{l-1},\dots,k_1} P(\tilde{d}_l = k_l|\tilde{d}_{l-1} = k_{l-1})P(\tilde{d}_{l-1} = k_{l-1}|\tilde{d}_{l-2} = k_{l-2}) \cdots P(\tilde{d}_2 = k_2|\tilde{d}_1 = k_1)P(\tilde{d}_1 = k_1|\tilde{d}_0 = j) \\
&= \sum_{k_l>0} \sum_{k_{l-1}:k_{l-1}>0} \sum_{k_{l-2},\dots,k_1} P(\tilde{d}_l = k_l|\tilde{d}_{l-1} = k_{l-1})P(\tilde{d}_{l-1} = k_{l-1}|\tilde{d}_{l-2} = k_{l-2}) \cdots P(\tilde{d}_1 = k_1|\tilde{d}_0 = j) \\
&\quad + \sum_{k_l>0} \sum_{k_{l-2},\dots,k_1} P(\tilde{d}_l = k_l|\tilde{d}_{l-1} = 0)P(\tilde{d}_{l-1} = 0|\tilde{d}_{l-2} = k_{l-2}) \cdots P(\tilde{d}_1 = k_1|\tilde{d}_0 = j) \\
& [\text{Since } k_l > 0 \text{ it follows by (A64) that } P(\tilde{d}_l = k_l|\tilde{d}_{l-1} = 0) = 0.] \\
&= \sum_{k_l>0} \sum_{k_{l-1}:k_{l-1}>0} \sum_{k_{l-2},\dots,k_1} P(\tilde{d}_l = k_l|\tilde{d}_{l-1} = k_{l-1})P(\tilde{d}_{l-1} = k_{l-1}|\tilde{d}_{l-2} = k_{l-2}) \cdots P(\tilde{d}_1 = k_1|\tilde{d}_0 = j) \\
& [\text{By iteration}] \\
&= \sum_{k_l>0} \sum_{k_{l-1},\dots,k_1:k_{l-1}>0,\dots,k_1>0} P(\tilde{d}_l = k_l|\tilde{d}_{l-1} = k_{l-1})P(\tilde{d}_{l-1} = k_{l-1}|\tilde{d}_{l-2} = k_{l-2}) \cdots P(\tilde{d}_1 = k_1|\tilde{d}_0 = j) \\
& [\text{By (A59)}] \\
&= \sum_{k_l>0} \sum_{k_{l-1},\dots,k_1:k_{l-1}>0,\dots,k_1>0} M_{k_l,k_{l-1}} \cdots M_{k_1,j} \\
& [\text{Since } k_l > 0,\ k_{l-1} > 0, \cdots, k_1 > 0, j > 0 \text{ if follows by (A61) that}] \\
&= \sum_{k_l>0} \sum_{k_{l-1},\dots,k_1:k_{l-1}>0,\dots,k_1>0} \tilde{M}_{k_l,k_{l-1}} \cdots \tilde{M}_{k_1,j} \\
& [\quad \tilde{M}_{k_l,k_{l-1}} \geq 0 \quad] \\
&\leq \sum_{k_l>0} \sum_{k_{l-1},\dots,k_1} \tilde{M}_{k_l,k_{l-1}} \cdots \tilde{M}_{k_1,j} \\
& [\text{By (A60)}] \\
&= \sum_{k_l>0} \sum_{k_{l-1},\dots,k_1} P(d_l = k_l|d_{l-1} = k_{l-1})P(d_{l-1} = k_{l-1}|d_{l-2} = k_{l-2}) \cdots P(d_1 = k_1|d_0 = j) \\
& [\text{By the Markovianity}] \\
&= \sum_{k_l>0} P(d_l = k_l|d_0 = j) \\
&= P(d_l > 0|d_0 = j).
\end{aligned} \tag{A65}$$

Consequently

$$\begin{aligned}
P(\tilde{d}_l = 0|\tilde{d}_0 = j) &= 1 - P(\tilde{d}_l > 0|\tilde{d}_0 = j) \\
&\geq 1 - P(d_l > 0|d_0 = j) \\
&= P(d_l \leq 0|d_0 = j), \quad j > 0.
\end{aligned} \tag{A66}$$

In the case $\tilde{d}_0 = 0$ we know that this is an absorbing state, and thus $P(\tilde{d}_l = 0|\tilde{d}_0 = 0) = 1$. Consequently, $P(d_l \leq 0|d_0 = 0) \leq P(\tilde{d}_l = 0|\tilde{d}_0 = 0) = 1$. This thus proves the inequality in (A56).

With the initial distribution (A57) we find

$$\begin{aligned} P(d_l \le 0) &= \sum_{j\in\mathbb{Z}} P(d_l \le 0 | d_0 = j) P(d_0 = j) \\ &= \sum_{j\ge 0} P(d_l \le 0 | d_0 = j) P(\tilde{d}_0 = j) \\ &\le \sum_{j\ge 0} P(\tilde{d}_l = 0 | \tilde{d}_0 = j) P(\tilde{d}_0 = j) \\ &= P(\tilde{d}_l = 0). \end{aligned} \tag{A67}$$

□

11. Bounds for Schöning Walks and Groverized Walks

Here we combine the previous observations in order to obtain the following lower bounds on the success-probability of the Schöning process. We also obtain the inequalites needed for determining the desired bound on the success-probability of the the Groverized walk.

Proposition 9. *Let $(x_l)_{l\in\mathbb{N}}$ be the Schöning process for bit strings of length n, with $x^\star$ the selected satisfying assignment. Let $(\tilde{d}_l)_l$ be the process as defined by (A5) and (A6). Let $(d_l)_l$ be the Markov chain as defined by the transition probabilites (A55) for $q = 2/3$ in Lemma 8, for the initial state*

$$P(d_0 = j) = \begin{cases} P(\tilde{d}_0 = j) = P(d_H(x_0, x^\star) = j) = \frac{1}{2^n}\binom{n}{j}, & n \ge j \ge 0 \\ 0, & \text{otherwise} \end{cases} \tag{A68}$$

Then

$$\begin{aligned} P(x_m = x^\star) &\ge P(\tilde{d}_m = 0) \\ &\ge P(d_m \le 0) \\ &= \sum_{\substack{j,l: j+m-2l\le 0,\\ 0\le j\le n,\\ 0\le l\le m}} \frac{1}{2^n}\binom{n}{j}\binom{m}{l}\left(\frac{1}{3}\right)^l\left(\frac{2}{3}\right)^{m-l} \end{aligned} \tag{A69}$$

and

$$\begin{aligned} P\big(x_m = x^\star \big| d_H(x_0|x^\star) = j\big) &\ge P(\tilde{d}_m = 0|\tilde{d}_0 = j) \\ &\ge P(d_m \le 0|d_0 = j) \\ &= \sum_{\substack{l: j+m-2l\le 0,\\ 0\le l\le m}} \binom{m}{l}\left(\frac{1}{3}\right)^l\left(\frac{2}{3}\right)^{m-l}. \end{aligned} \tag{A70}$$

Proof. By (A43) in Lemma 7 yields the first inequality in (A69).

By Proposition 6 we know that $(d_l)_l$ is a Markov chain with transition probabilities as in (A33) and initial distribution (A35). By these observations, it follows that the second inequality in (A69) is a direct application of (A58) in Lemma 8. By Lemma 8, we also know that the Markov chain $(d_l)_l$ defined by the transition probabilities

$$P(d_{l+1} = j | d_l = k) = \frac{1}{3}\delta_{j,k-1} + \frac{2}{3}\delta_{j,k+1}, \quad \forall j,k \in \mathbb{Z}, \quad \forall l \in \mathbb{N}. \tag{A71}$$

and initial distribution

$$P(d_0 = j) = \begin{cases} P(\tilde{d}_0 = j), & j \ge 0 \\ 0, & j < 0 \end{cases} = \begin{cases} \frac{1}{2^n}\binom{n}{j}, & 0 \le j \le n \\ 0, & \text{otherwise} \end{cases} \tag{A72}$$

From (A71) it follows that

$$P(d_m \le 0 | d_0 = j) = \sum_{l: j+m-2l\le 0,\, 0\le l\le m} \binom{m}{l}\left(\frac{1}{3}\right)^l\left(\frac{2}{3}\right)^{m-l} \tag{A73}$$

and thus

$$
\begin{aligned}
P(d_m \leq 0) &= \sum_j P(d_m \leq 0|d_0 = j)P(d_0 = j) \\
&\text{[By (A72)]} \\
&= \sum_{\substack{j,l:j+m-2l\leq 0,\\ 0\leq j\leq n,\\ 0<l<m}} \frac{1}{2^n}\binom{n}{j}\binom{m}{l}\left(\frac{1}{3}\right)^l\left(\frac{2}{3}\right)^{m-l}.
\end{aligned}
\tag{A74}
$$

Next, we turn to the inequalities in (A70). Inequality (A44) in Lemma 7 yields the first inequality in (A70). The second inequality in (A70) is a direct application of (A56) in Lemma (8). By (A73) we already know the last equality in (A70). □

12. Relation to the leading order analysis of the Schöning and GW process

Here we connect to the analysis of the asymptotic scaling in the main text, by obtaining the starting points, so to speak, of the leading order analysis of the Schöning process and the GW process.

a. Schöning process

For the Schöning process, the average number of repetitions needed to find a solution is given by

$$N_{\text{Schöning}} = \frac{1}{P(x_m = x^\star)}. \tag{A75}$$

By sequences of lower bounds on the ideal success-probability $P(x_m = x^\star)$, we thus obtain upper bounds on $N_{\text{Schöning}}$. The step from the true Schöning process to the walk on Z corresponds to one such inequality, i.e. to

$$P(x_m = x^\star) \geq P(d_m \leq 0) \tag{A76}$$

in (A69) in Proposition (9). The leading-order analysis in the main text is based on further such inequalities, with the rationale that the 'loss' of probability weight becomes irrelevant for the rates $\gamma = \lim_{n\to\infty}\frac{1}{n}\log N_{\text{Schöning}}$, if the inequalities are chosen to correspond to the leading order contributions. As a first step along these lines, we restrict to an event where we not only reach the desired solution, but also start the system x_0 at Hamming distance $d_H(x_0, x^\star) = j$. Trivially,

$$P(d_m \leq 0) \geq P(d_m \leq 0, d_0 = j) = P(d_0 = j)P(d_m \leq 0|d_0 = j), \tag{A77}$$

where can identify $P(d_0 = j)$ with $P(E_1)$ in the main text, i.e.

$$P(d_0 = j) = P(E_1) = \frac{1}{2^n}\binom{n}{\kappa n}, \quad j = \kappa n. \tag{A78}$$

Next, we wish to connect the remaining factor in (A77), i.e., $P(d_m \leq 0|d_0 = j)$, to the probability $P(E_2)$, which we recall from the main text corresponds to the event E_2, where precisely νm steps decrease the Hamming distance, while precisely $(1-\nu)m$ steps increase the Hamming distance. (For the walk on $\mathbb{Z}$ this extends to νm steps in the negative direction, and $(1-\nu)m$ steps in the positive direction.) We conclude that the total decrease is

$$d_0 - d_m = (2\nu - 1)m. \tag{A79}$$

Let us also recall that the combination of E_1 and E_2 is successful, i.e. leads to $d_m \leq 0$, if

$$(2\nu - 1)m \geq \kappa n. \tag{A80}$$

It is useful to note that $(d_l)_l$ is not only Markovian, but also translation symmetric, which means that the change $d_0 - d_m$ is independent of the initial state d_0, i.e., the joint distribution of these factorize. (As a side remark, this independence also means that $P(E_1 \cap E_2) = P(E_1)P(E_2)$.) Hence,

$$
\begin{aligned}
&P(d_m \leq 0|d_0 = \kappa n) \\
&= P(d_0 - d_m \geq \kappa n|d_0 = \kappa n) \\
&\text{[Since } d_0 - d_m \text{ is independent of } d_0] \\
&= P(d_0 - d_m \geq \kappa n).
\end{aligned}
\tag{A81}
$$

By comparison of (A81) with (A79) it follows that

$$\begin{aligned} &P(d_m \leq 0|d_0 = \kappa n) \\ &= P(d_0 - d_m \geq \kappa n) \\ &\geq P(E_2), \quad \text{if} \quad (2\nu - 1)m \geq \kappa n. \end{aligned} \tag{A82}$$

Alternatively, we can reach the same conclusion by comparing (7) with (A73) to see that

$$P(E_2) = \binom{m}{\nu m} \left(\frac{1}{3}\right)^{\nu m} \left(\frac{2}{3}\right)^{(1-\nu)m} \leq P(d_m \leq 0|d_0 = \kappa n), \quad \text{if} \quad (2\nu - 1)m \geq \kappa n \tag{A83}$$

By (A75), (A76), (A77), (A78) and (A82), we can conclude that

$$N_{\text{Schöning}} \leq \frac{1}{P(E_1)P(E_2)}, \quad \text{if} \quad (2\nu - 1)m \geq \kappa n. \tag{A84}$$

b. GW process

For the GW process, let us recall that it consists of a classical outer loop that at each round assigns a definite (classical) initial state, while the walk-process is Groverized. We assume that the number of iterations of the Grover-procedure is tuned to the density of successful walks, for a specific initial Hamming distance $j = \kappa n$, i.e., to the success probability $P\big(x_m = x^\star \big| d_H(x_0|x^\star) = \kappa n\big)$. In the analysis we lower bound the success-probability by assuming that process fails whenever $d_H(x_0, x^\star) \neq \kappa n$ (which may be pessimistic). The probability to obtain the initial state x_0 with Hamming distance κn is $P\big(d_H(x_0, x^\star) = \kappa n\big)$, and thus in average we need to repeat the outer loop $1/P\big(d_H(x_0, x^\star) = \kappa n\big)$ times to be guaranteed to at least once reach the initial Hamming distance κn. In the successful case, the Grover procedure requires $1/\sqrt{P\big(x_m = x^\star \big| d_H(x_0|x^\star) = \kappa n\big)}$ iterations. Consequently, an upper bound on the total number of steps is

$$\begin{aligned} N_{\text{GW}} &\leq \frac{1}{P\big(d_H(x_0, x^\star) = \kappa n\big)\sqrt{P\big(x_m = x^\star \big| d_H(x_0|x^\star) = \kappa n\big)}} \\ &\quad \text{[By Proposition (9)]} \\ &\leq \frac{1}{P(d_0 = \kappa n)\sqrt{P(d_m \leq 0|d_0 = \kappa n)}} \\ &\quad \text{[By (A78) and (A82)]} \\ &\leq \frac{1}{P(E_1)\sqrt{P(E_2)}} \quad \text{if} \quad (2\nu - 1)m \geq \kappa n. \end{aligned} \tag{A85}$$