Available online at www.sciencedirect.com

SciVerse ScienceDirect

Stochastic Processes and their Applications 121 (2011) 2455–2473

stochastic processes and their applications

www.elsevier.com/locate/spa

DNA approach to scenery reconstruction

Heinrich Matzinger[a,*], Angelica Pachon Pinzon[b]

[a] School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332, United States
[b] Centre for Discrete Mathematics and its Applications DIMAP, University of Warwick, Coventry CV4 7AL, United Kingdom

Received 1 October 2009; received in revised form 12 April 2011; accepted 27 April 2011
Available online 30 May 2011

Abstract

The basic reconstruction problem lead with the general task of retrieving a scenery from observations made by a random walker. A critical factor associated with the problem is reconstructing the scenery in polynomial time. In this article, we propose a novel technique based on the modern DNA sequencing method for reconstructing a 3-color scenery of length n. The idea is first to reconstruct small pieces of length order log n and then assembled them together to form the required piece. We show that this reconstruction and assembly for a finite piece of a 3-color scenery takes polynomial amount of time.

1. Introduction

Scenery reconstruction considers a random walk moving around in a landscape or scenery ξ, producing a sequence of observations. The problem is to retrieve the scenery ξ based on a given sequence of observations χ. More specifically, consider a coloring of the integers $\xi : \mathbb{Z} \to \{0, 1, 2, \dots, C - 1\}$ which we shall call a scenery (i.e. random media). Let S be a recurrent random walk starting at the origin. We assume that we observe the scenery along the path of the random walk S, that is, we observe the color $\chi_t := \xi(S_t)$ at time t. The *scenery reconstruction* problem investigates whether one can identify a coloring of the integers, using only the color record seen along a random walk path. In other words, if one path realization of χ a.s. determines ξ. In [9], it has been proven that almost every 2-color scenery can be reconstructed when seen along the path of a simple symmetric random walk. The scenery is taken as i.i.d. with equiprobable symbols.

[*] Corresponding author. Tel.: +1 6783275755.
E-mail addresses: matzi@math.gatech.edu (H. Matzinger), A.Y.Pachon-Pinzon@warwick.ac.uk (A.P. Pinzon).

0304-4149/$ - see front matter © 2011 Published by Elsevier B.V.
doi:10.1016/j.spa.2011.04.010

It is important to note that in a scenery with 2 instead of 3 colors, the reconstruction problem becomes entirely different, more of statistical in nature than a combinatorial problem. The techniques for solving the reconstruction problem when the random walk is not skip-free [3,4,7] or two dimensional [6] are different to the approach in [9] for solving the reconstruction problem with 2-colors by observing it along a simple random walk path. Approaches for reconstruction with error-corrupted observations have been introduced in [2,10]. In [14], a continuous version of the problem is treated.

In [11–13], it has been proven that in certain cases finite pieces of a 3-color sceneries close to the origin can be reconstructed in polynomial time. However, the complex theoretical nature of these algorithms makes it difficult for scientists to implement them using a computer program.

In this article, a novel practical algorithm is presented for the reconstruction of a finite piece of scenery around the origin, which works in polynomial time. This is significant, as the existing algorithms are more of theoretical interest and too difficult to implement even using computers.

The algorithm is based on the same idea as used by one of the techniques for DNA reconstruction known as "polymerase chain reaction (PCR)", developed by Kary Mullis in 1983. PCR is a scientific technique in molecular biology to amplify a single or a few copies of a piece of DNA across several orders of magnitude, generating thousands to millions of copies of a particular DNA sequence. In this paper, we only explain how our algorithm works mathematically. For the biological and conceptual details of PCR, see [1].

The algorithm first obtains micro-strings from a finite string of ξ and then assembles them. The micro-strings are of logarithmic order in the length of the piece to be reconstructed. The algorithm reconstructing the micro-strings needs exponential time in the size of the micro-strings, while, exponential of logarithmic leads to amount of time polynomial!

In [5], it has been showed that there are sceneries which cannot be reconstructed, thus, only normally typical sceneries are considered in scenery reconstruction. Therefore, we assume that the scenery is itself the outcome of a random process and one tries to show that almost all sceneries can be reconstructed up to equivalence, where ξ and $\bar{\xi}$ are called equivalent if there exists an $a \in \mathbb{Z}$ such that for all $z \in \mathbb{Z}$,

$$\xi(z) = \bar{\xi}(z+a) \quad \text{or}$$
$$\xi(z) = \bar{\xi}(-z).$$

In other words, two sceneries are equivalent if one of them is obtained from the other by either translation, reflection or their simultaneous composition.

Thus, most results so far concern this question if an infinite amount of observations $a.s.$ determines the whole scenery up to reflection and translation.

In this article, we show that with high probability a piece of length order n can be reconstructed in finite time. We say that a sequence of events A_n holds with high probability (w.h.p.) if $\lim_{n\to\infty} P(A_n) = 1$.

1.1. Notation and results

Assume that the scenery $\{\xi_i\}_{i\in\mathbb{Z}}$ is a double infinite sequence of i.i.d. random variables with state space $\{0, 1, 2\}$ such that

$$P(\xi_i = 0) = P(\xi_i = 1) = P(\xi_i = 2) = 1/3.$$

Then, ξ will designate a path realization of $\{\xi_i\}_{i\in\mathbb{Z}}$. Consider $\{S_t\}_{t\in\mathbb{N}}$ to be a simple symmetric random walk starting at the origin and χ_t be the observation of the random walk at time t, i.e.

$$\chi_t := \xi(S_t).$$

Now, we formulate our main result. Let $\mathcal{A}(.)$ be a map which represents an algorithm that takes the first $\mathcal{T}^n$ observations of χ as input and produces a piece of scenery as output. The main result of this paper is that w.h.p., the reconstructed piece contains the restriction of ξ to $[-n, n]$, and is contained in the restriction of ξ to $[-4n, 4n]$. More precisely, we have the following theorem.

Theorem 1.1. *For every $n \in \mathbb{N}$ large enough, there exists a map*

$$\mathcal{A} : \{0, 1, 2\}^{\mathcal{T}^n} \to \cup_{m\in[2n,8n]}\{0, 1, 2\}^m$$

such that

$$P\begin{pmatrix} \exists i_1, i_2 \text{ such that} \\ i_1 \in [-4n, -n];\ i_2 \in [n, 4n] \text{ and} \\ (\mathcal{A}(\chi_1\chi_2 \dots \chi_{\mathcal{T}^n}) = \xi_{i_1}\xi_{i_1+1}\dots\xi_{i_2} \text{ or} \\ \mathcal{A}(\chi_1\chi_2 \dots \chi_{\mathcal{T}}) = \xi_{i_2}\xi_{i_2-1}\dots\xi_{i_1}) \end{pmatrix} \geq 1 - n^{-\beta}, \tag{1.1}$$

where, $\mathcal{T}^n := n^6 + n^{9k_3+9}$ whilst $k_3 > 0$ and $\beta > 0$ are constants not depending on n.

2. Proof of Theorem 1.1

2.1. Main ideas

Reconstructing from pieces

Definition 2.1. Let $s = s_1s_2\dots s_i$ and $r = r_1r_2\dots r_j$ be two strings where $i < j$. The transpose of s is denoted by $s^* := s_is_{i-1}\dots s_1$.

We say that s appears in more than one location in r if and only if there exists $x + i - 1, y + i - 1 \leq j$ with $x \neq y$, such that at least one of the following three conditions hold:

1. $s = r_xr_{x+1}\dots r_{x+i-1}$ and $s = r_yr_{y+1}\dots r_{y+i-1}$
2. $s = r_xr_{x+1}\dots r_{x+i-1}$ and $s^* = r_yr_{y+1}\dots r_{y+i-1}$
3. $s^* = r_xr_{x+1}\dots r_{x+i-1}$ and $s^* = r_yr_{y+1}\dots r_{y+i-1}$.

The idea of reconstruction from pieces converts the problem of reconstructing a string of scenery of length order n, to reconstruct short substrings. This is important since short substrings can be reconstructed much quicker.

Let $\mathcal{S}$ be a string of length order l. To reconstruct $\mathcal{S}$ using substrings, one first needs to show that with high probability in an i.i.d. 3-equiprobable-color string of length order l, every substring of length $k \ln l$ appear only in one location, with $k > 0$ a constant large enough but not depending on l. The assembling works as follows:

1. Assume we have a collection W of substrings of the string $\mathcal{S}$, and that for each substring s of $\mathcal{S}$ of length $k \ln l + 1$ there exists $w \in W$ such that s is a substring of w.
2. Assemble the substrings in W (or their transpose) by checking if they overlap at least $k \ln l$ consecutive letters.

Consider the following example.

Example 2.1. Assume we are given the words

$$w_1 = 22321, \qquad w_2 = 3212, \qquad w_3 = 1212.$$

Assume that these three words are all substrings of a string $\mathcal{S}$ in which every 3-letter substring appears at most in one location. (A substring appears in the string when we read it from left to right or right to left.) Then we assemble w_1, w_2 and w_3 in order to get a bigger substring of $\mathcal{S}$.

First take w_1 and w_2, and see on which three letter group they coincide.

$$\begin{array}{lcccccc}
w_1 & 2 & 2 & 3 & 2 & 1 & \\
w_2 & & & 3 & 2 & 1 & 2 \\
\hline
w_4 & 2 & 2 & 3 & 2 & 1 & 2
\end{array}$$

Now puzzle together w_4 and the transpose $w_3^t = 2121$.

$$\begin{array}{lccccccc}
w_4 & 2 & 2 & 3 & 2 & 1 & 2 & \\
w_3^t & & & & 2 & 1 & 2 & 1 \\
\hline
w_5 & 2 & 2 & 3 & 2 & 1 & 2 & 1
\end{array}$$

The assembled string $w_5 = 2232121$ must be a substring of $\mathcal{S}$.

Reconstructing a substring

For the reconstruction from pieces it is assumed that a collection of substrings of the string which we want to reconstruct is given. Stopping times are used for the production of these substrings, as follows.

Assume x and y are two non-random integer numbers such that $x < y$. The aim here is to reconstruct the "substring" written between x and y, i.e. we would like to reconstruct $\xi_x \xi_{x+1} \dots \xi_y$. Assume that we have the observations $\chi_1 \chi_2 \dots$ and the corresponding times when the random walk S visits x or y.

Let ν_i be the ith visit of the random walk to x and τ_i be the ith visit to y. Then, when the random walk crosses from x to y in the shortest period of time, we have the random walk, which only takes steps to the right. Hence, during such a minimal time in the observations we are seeing a copy of the substring $\xi_x \xi_{x+1} \dots \xi_y$.

Given that our random walk is recurrent, it will cross in the shortest period of time from x to y infinitely often. Hence, to reconstruct the substring between x and y take

$$\chi_{\nu_i} \chi_{\nu_i+1} \chi_{\nu_i+2} \cdots \chi_{\tau_j}$$

where ν_i and τ_j satisfy

$$\tau_j - \nu_i = \min_{k,l} \{|\tau_k - \nu_l|\}.$$

Now, a priori the stopping times τ_j and ν_i are not observable. In the next subsection we explain how often we can figure them out solely based on the observations χ.

Representation of the scenery on a 3-regular tree

To reconstruct the micro-strings, they will be using a representation of the scenery ξ on a 3-regular tree $T = (E_T, V_T)$ equipped with a non-random coloring $\psi : V_T \to \{0, 1, 2\}$. This idea was introduced in [8]. Formally, the idea is as follows.

Let $T = (E_T, V_T)$ be a 3-regular tree with root v_0, and $\psi : V_T \to \{0, 1, 2\}$ be a (random) coloring on T such that every vertex $v \in V_T$ has its 3-adjacent vertices colored in three different colors 0, 1 and 2, i.e.

$$\forall v \in V_T, \quad \{\psi(w) | w \in \{v_1, v_2, v_3\}\} = \{0, 1, 2\}, \tag{2.1}$$

where v_1, v_2, v_3 are the three vertices adjacent to v.

Let ψ^0, ψ^1 and ψ^2 be three non-random colorations such that each one satisfies the condition (2.1) and $\psi^i(v_0) = i$ for $i = 0, 1, 2$. We assume that ψ is always equal to either ψ^0, ψ^1 or ψ^2. When $\xi(0) = 0$, then $\psi = \psi^0$, whilst when $\xi(0) = 1$, then $\psi = \psi^1$ and finally $\xi(0) = 2$ implies $\psi = \psi^2$.

So the color at the origin of ψ is the same as that at the origin of ξ. Also, ψ "is only random as far as $\xi(0)$ is".

We call the map $R : I \cap \mathbb{Z} \to V_T$ (where I is an interval) on T, a nearest neighbor path on T, if and only if for all $z \in I \cap \mathbb{Z}$: $R(z)$ and $R(z+1)$ are adjacent vertices, i.e.

$$\forall z \in I \cap \mathbb{Z}, \quad \{R(z), R(z+1)\} \in E_T.$$

Let $\zeta : I \cap \mathbb{Z} \to \{0, 1, 2\}$ be a 3-color scenery on $I \cap \mathbb{Z}$, then we can say that R generates ζ on ψ if and only if $\zeta = \psi \circ R$.

In order to represent the double infinite sequence ξ as a nearest neighbor walk R on a colored tree ψ we need a uniqueness condition.

Proposition 2.1. *Let $\mathcal{S} = s_1 s_2 \ldots s_j \in \{0, 1, 2\}^j$ be a string, and v be a vertex in V_T such that $\psi(v) = s_1$, then there exists a unique nearest neighbor path $\{R(t)\}_{t \in [1,j]}$ on T such that,*

$$\psi(R(1)) \ldots \psi(R(j)) = s_1 s_2 \ldots s_j,$$

with $R(1) = v$. Thus, R generates $\mathcal{S}$.

Proof. Suppose that $\{U(t)\}_{t \in [1,j]}$ and $\{W(t)\}_{t \in [1,j]}$ are two nearest neighbor paths such that

$$\psi(U(1)) \ldots \psi(U(j)) = s_1 s_2 \ldots s_j = \psi(W(1)) \ldots \psi(W(j)),$$

with $U(1) = W(1) = v$. Then, at time 2, $U(2)$ and $W(2)$ will be over an adjacent vertex from v, and also $\psi(U(2)) = s_2 = \psi(W(2))$, so from (2.1) $U(2) = W(2)$. Suppose that at time $k < j$, $U(k) = W(k) = v_k$, then at time $k+1$, $U(k+1)$ and $W(k+1)$ will be over an adjacent vertex from v_k, and also $\psi(U(k+1)) = s_{k+1} = \psi(W(k+1))$, again from (2.1) $U(k+1) = W(k+1)$. Thus, by the induction argument we have $\{U(t)\}_{t \in [2,j]} = \{W(t)\}_{t \in [2,j]}$. □

Proposition 2.1 states that, given any sequence of colors $\mathcal{S}$, there exists a unique nearest neighbor walk that generates $\mathcal{S}$ once we know where it starts.

Note that the representation of ξ, as a nearest neighbor path R, defines a simple random walk on the graph (E_T, V_T):

More precisely, for $z \geq 0$, we have $P(\{R(z+1) = v_i | R(z) = v\}) = 1/3$, with $v_i, i = \{1, 2, 3\}$, designating the 3-adjacent vertices of v. Thus $\{R(z)\}_{z \in \mathbb{N}}$ is a random walk on our tree (E_T, V_T) starting at the origin. Same thing for $\{R(-z)\}_{z \in \mathbb{N}}$. Let R designate the unique nearest neighbor path R on T with $R(0) = v_0$ such that

$$\psi(R(z)) = \xi(z),$$

$\forall z \in \mathbb{Z}$. We call $\{R(z)\}_{z \in \mathbb{Z}}$ the representation of the scenery $\{\xi_i\}_{i \in \mathbb{Z}}$ as nearest neighbor walk on T.

Using this representation, our problem of reconstructing ξ is translated to reconstructing R using the observations χ. Though we do not know R yet, we can easily figure out $R \circ S$ just with the observations χ.

By definition we have $\psi(R(z)) = \xi(z)$, then, $\forall t \in \mathbb{N}$

$$\psi(R(S_t)) = \xi(S_t) = \chi_t$$

hence,

$$\psi \circ (R \circ S) = \chi.$$

Note that $R \circ S$ is also a nearest neighbor walk on T, and it is the only one, which generates χ on ψ.

In this order of ideas, if we know R, we would also know ξ, so the problem of reconstructing ξ is equivalent to reconstructing R given $R \circ S$. Let us look at an example:

Example 2.2. Suppose that the scenery is

$$\xi_z = 0201001\ldots$$
$$z = 0123456\ldots$$

and S_t represents a random walk, which produces χ_t, then we have

$$S(0), \ldots, S(10) = 0, 1, 2, 1, 2, 3, 4, 3, 4, 5, 6 \quad \text{and}$$
$$\chi(0), \ldots, \chi(10) = 0, 2, 0, 2, 0, 1, 0, 1, 0, 0, 1.$$

Now observe R and $R \circ S$ over the tree (Fig. 1). They are respectively the representations of ξ and χ.

Now, let v and w be two different vertices of V_T visited by R, with $\{x_j, x_{j-1}, \ldots, x_1\}$ and $\{y_1, y_2, \ldots, y_i\}$ representing the set of all times when R visited v and w respectively.

In infinite time, R is transient (see Lemma 5 in [8]), then $a.s.$ every vertex of V_T is visited only a finite number of times. It implies that if two vertices v and w are far enough from each other on the tree, with v closer to the origin than w, then the last visit to v occurs before the first visit to w. Thus, in finite time if v is far enough from w the following condition holds w.h.p.

$$x_j < x_{j-1} < \cdots < x_2 < x_1 < y_1 < y_2 < \cdots < y_i,$$

so in finite time it is possible to get the shortest paths between v and w. Thus, by the method of "reconstructing a substring", there is a simple way to reconstruct the scenery between x_1 and y_1. *Take the pair of times* (t, s) *minimizing* $(s - t)$ *under the constraints*

$$R(S_t) = v \quad \text{and} \quad R(S_s) = w, \quad s > t.$$

Then, $a.s.$ the observations during the interval of time $[t, s]$ are equal to the scenery between x_1 and y_1, i.e.

$$\chi_t \chi_{t+1} \cdots \chi_s = \xi_{x_1} \xi_{x_1+1} \cdots \xi_{y_1}.$$

The reason why this works is because the random walk, when going in shortest time from x_1 to y_1, has a straight path. This also reveals the piece of scenery ξ between x_1 and y_1. Also, we know that the recurrent random walk will pass from x_1 to y_1 in the shortest period of time infinitely often.

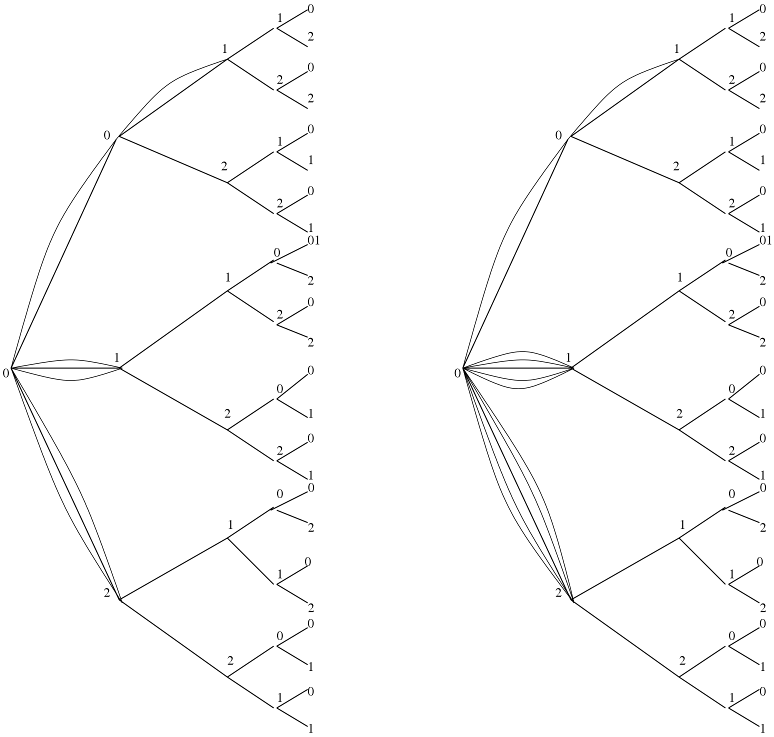

Fig. 1. Left and right trees, represent R and $R \circ S$ respectively.

2.2. Reconstruction algorithm

Let us describe the algorithm to reconstruct a piece of ξ which contains the substring $\xi(-n)\xi(-n+1)\xi(-n+2)\dots\xi(n)$ contained in the string $\xi(-4n)\xi(-4n+1)\xi(-4n+2)\dots\xi(4n)$.

Let $\mathcal{T}^n := n^6 + n^{9k_3+9}$, where $k_3 > 0$ is a constant not depending on n and which will be defined subsequently. Our algorithm only takes as input the observations up to time $\mathcal{T}^n$. Hence, the algorithm only uses a polynomial number of observations in the length of the piece of scenery to be reconstructed. Let V^n denote the subset of V_T containing those vertices which have been visited by $R \circ S$ up to time $\mathcal{T}^n$ and are not further away from the root than n, i.e.

$$V^n := \{R(S(t)) \mid d(R(S(t)), v_0) \leq n, t \in [0, \mathcal{T}^n]\}.$$

1. Determine V^n.
2. Build a "lexical" W of words which can be obtained by shortest path: for any pair $(v_1, v_2) \in (V^n) \times (V^n)$ such that up to time $\mathcal{T}^n$, the nearest neighbor walk $R \circ S$ goes at least once from v_1 to v_2 in at most $(k + 2k_1)\ln n + 1$ steps:
   take (t, s) minimizing $s - t$ under the constraints
   $$R(S_t) = v_1 \quad \text{and} \quad R(S_s) = v_2, \quad s > t;\ s, t \leq \mathcal{T}^n.$$
   The string $\chi_t \chi_{t+1} \dots \chi_s$ is one of the reconstructed words. Now only keep those words with length at least $k \ln n + 1$. The set of words obtained in this way is denoted by W.

3. Assemble the words from W. For this use the following assembling rule: in order to "puzzle two words together", the words or their transposes must coincide on at least $k \ln n$ contiguous letters. Start with a word which was obtained using the vertex v_0. Assemble one word after the other to the already assembled word. Produce in this manner one piece of scenery. (Some words might not be used since they might occur in another interval which is not connected to the reconstructed interval.)

Now we need to prove that this reconstruction algorithm works w.h.p. Let $\mathcal{A}lg$ be the event on which our reconstruction algorithm works. More precisely,

$$\mathcal{A}lg := \left\{ \begin{matrix} \text{the reconstructed piece contains } \xi(-n)\xi(-n+1)\xi(-n+2)\dots\xi(n), \\ \text{and is contained in the string } \xi(-4n)\xi(-4n+1)\xi(-4n+2)\dots\xi(4n) \end{matrix} \right\}.$$

Defining the following events:

$$B_1 := \left\{ \{ \forall z \notin [-4n, 4n], \quad d(R(z), v_0) > n \} \right\},$$

$$B_2 := \left\{ \begin{matrix} \text{in the string } \xi_{-4n}\xi_{-4n+1}\dots\xi_{4n}, \\ \text{there is no word of length } k \log n \text{ appearing} \\ \text{in two different places} \end{matrix} \right\},$$

$$B_3 := \left\{ \begin{matrix} \text{for every } z \in [-4n, 4n], \\ R(z - k_1 \ln n) \notin R[z, +\infty] \end{matrix} \right\},$$

$$B_4 := \left\{ \begin{matrix} \text{for every } z \in [-4n, 4n], \\ R(z + (k + k_1)\ln n + 1) \notin R[-\infty, z + k \ln n + 1] \end{matrix} \right\},$$

$$B_5 := \left\{ \begin{matrix} \text{every subinterval of } [-4n, 4n] \text{ of length } (k + 2k_1)\ln n + 1, \\ \text{gets crossed in a straight manner by } S \text{ before time } n^{9k_3+9} \end{matrix} \right\},$$

$$B_6 := \left\{ \{ R(z) \neq v_0, \quad \forall z \notin [-n/2, n/2] \} \right\},$$

$$B_7 := \left\{ \{ \forall z \in [n - k_1 \ln n, n + k_1 \ln n], \quad d(R(z), v_0) \leq n \} \right\},$$

the main combinatorial lemma is as follows.

Lemma 2.1. *We have*

$$B_1 \cap B_2 \cap B_3 \cap B_4 \cap B_5 \cap B_6 \cap B_7 \subset \mathcal{A}lg.$$

Proof. Let (v_1, v_2) be a pair of vertices of V^n that is selected by our algorithm and leads to a reconstructed word w for W. Let (t, s) be the time pair minimizing $s - t$ under the constraint $R(S_t) = v_1$ and $R(S_s) = v_2$ whilst $s > t$ and $s, t < \mathcal{T}^n$. Hence, the reconstructed word w is equal to the observations χ during the time interval $[t, s]$, i.e.,

$$w = \chi_t \chi_{t+1} \cdots \chi_s.$$

Because of B_1, we have that $R(z)$ can only be in V^n, when $z \in [-4n, 4n]$. It follows that if B_1 holds, and as $v_1, v_2 \in V^n$, we must have $S_s, S_t \in [-4n, 4n]$. Denote S_t by z_1 and S_s by z_2. The algorithm chooses only pairs (v_1, v_2) for which the nearest neighbor walk $R \circ S$ goes in less than $(k + 2k_1)\ln n + 1$ steps from one to the other. It follows, that

$$|z_1 - z_2| \leq (k + 2k_1)\ln n + 1.$$

We have already seen that $z_1, z_2 \in [-4n, 4n]$. But according to the event B_5, every interval of $[-4n, 4n]$ of length less or equal to $(k+2k_1)\ln n+1$ gets crossed in a straight manner by S before time $\mathcal{T}$. This implies that before time $\mathcal{T}^n$, the random walk S will walk in a straight manner from z_1 to z_2. Hence, if (t, s) is to minimize $s - t$ under the constraint $R(S_t) = v_1$ and $R(S_s) = v_2$; and $s - t > 0$ with $s, t \leq \mathcal{T}^n$, then, necessarily during the time (t, s) the random walk S must be a straight way from z_1 to z_2. (Otherwise, we would not have a minimum, since the straight walk would be shorter). Since, during the time interval (t, s) the random walk makes steps only in one direction, the observations during that time are a copy of the scenery between the points z_1 and z_2. More precisely, assume that the random walk S makes only steps to the right during the time interval (t, s). Then the observations $\chi_t \chi_{t+1} \dots \chi_s$ are equal to $\xi_{z_1}\xi_{z_1+1}\xi_{z_1+2}\dots\xi_{z_2}$. Hence the reconstructed word w, is equal to $\xi_{z_1}\xi_{z_1+1}\xi_{z_1+2}\dots\xi_{z_2}$ and is part of the scenery ξ restricted to $[-4n, 4n]$. The same conclusion holds true if the steps taken during the time (s, t) are all to the left, but then the reconstructed word is equal to $\xi_{z_2}\xi_{z_2-1}\xi_{z_2-2}\dots\xi_{z_1}$. We have just proved that if B_1 and B_5 both hold, then the collection of words W reconstructed by our algorithm contains only words contained in the part of the scenery: $\xi_{-4n}\xi_{-4n+1}\xi_{-4n+2}\dots\xi_{4n-1}\xi_{4n}$.

We have so far assumed that there are no "wrong words" in W. For the algorithm to puzzle together the words for desired reconstructed piece, enough good words in W should be ensured. This is what we are going to check next.

Let z and $z + k\ln n + 1$ be in $[-n, n]$, with

$$v_1 := R(z - k_1 \ln n)$$

and

$$v_2 := R(z + k\ln n + 1 + k_1 \ln n).$$

Let z_1 be the largest $m \in \mathbb{Z}$ for which $R(z) = v_1$. Then because of the event B_3, we have $z_1 < z$ and hence

$$z_1 \in [z - k_1 \ln n, z].$$

Let z_2 be the smallest $m \in \mathbb{Z}$ such that $R(z) = v_2$. Because of B_4, we find that $z_2 > z + k\ln n + 1$ and hence

$$z_2 \in [z + k\ln n + 1, z + k\ln n + 1 + k_1 \ln n].$$

So, the couple (z_1, z_2) minimizes $|z_1 - z_2|$ under the constraint

$$R(z_1) = v_1, \qquad R(z_2) = v_2.$$

By the event B_5 up to time $\mathcal{T}^n$, the random walk passes from v_1 to v_2 at least once in a straight manner. Let the times of such a straight crossing be denoted by (s_1, s_2), hence $S_{s_1} = v_1$ and $S_{s_2} = v_2$ and during the time interval (s_1, s_2), the random walk S takes steps only in one direction and $s_1, s_2 \leq \mathcal{T}^n$. Since z_1, z_2 minimizes $|z_1 - z_2|$ under $R(z_1) = v_1$ and $R(z_2) = v_2$, we have the shortest path for the nearest neighbor walk $R \circ S$ from v_1 to v_2 taking $|z_1 - z_2|$ steps, and this can only occur when S walks straight from z_1 to z_2. Hence, the time (s_1, s_2) corresponds to a straight crossing of the random walk S from z_1 to z_2, so that:

$$\chi_{s_1}\chi_{s_1+1}\dots\chi_{s_2} = \xi_{z_1}\xi_{z_1+1}\dots\xi_{z_2}.$$

Again, note that up to time $\mathcal{T}^n$, the time pair (s_1, s_2) minimizes $|s_2 - s_1|$ under the constraint $R(S_{s_1}) = v_1$ and $R(S_{s_2}) = v_2$. So, as soon as (v_1, v_2) gets picked by our algorithm, then

$$\xi_{z_1}\xi_{z_1+1}\dots\xi_{z_2}$$

will be a reconstructed word by our algorithm in the collection W. Now, we know that $|s_1 - s_2|$ are less apart than $(k + 2k_1)\ln n + 1$. This implies that the nearest neighbor walk $R \circ S$ goes from v_1 to v_2 in at most to $(k+2k_1)\ln n + 1$ steps. This is the first criteria for the pair of vertices (v_1, v_2) to get selected. The second criteria is that $v_1, v_2 \in V^n$, this is guaranteed by the event B_7. We have just proved that if all the events B_3, B_4, B_5 and B_7 hold, then the substring

$$\xi_{z_1}\xi_{z_1+1}\dots\xi_{z_2} \tag{2.2}$$

is obtained by our reconstruction algorithm and added to W. The piece of scenery (2.2), contains the piece

$$w_z := \xi_z\xi_{z+1}\dots\xi_{z+k\ln n+1}.$$

So, we have proved that for every interval

$$[z, z + k\ln n + 1] \subset [n, -n],$$

at least one word w containing the piece w_z, is in the set of words W.

We have proved that if B_1 and B_5 both hold, then the collection of words W reconstructed by our algorithm contains only words contained in the part of the scenery:

$$\xi_{-4n}\xi_{-4n+1}\xi_{-4n+2}\dots\xi_{4n-1}\xi_{4n}.$$

The event B_2 guarantees that the algorithm "puzzles" words of W together correctly, and the result is again a piece of

$$\xi_{-4n}\xi_{-4n+1}\xi_{-4n+2}\dots\xi_{4n-1}\xi_{4n}.$$

The algorithm starts puzzling with a word w_0 which was obtained using the vertex v_0. By B_6, the word w_0 is part of the restriction of ξ to $[-n, n]$. For every $[z, z+k\ln n+1]$ in $[-n, n]$, we have at least one word in W containing $\xi_z\xi_{z+1}\dots\xi_{z+k\ln n+1}$. This implies that the final reconstructed piece by our algorithm must contain the restriction of ξ to $[-n, n]$. It must also be contained in the restriction of ξ to $[-4n, 4n]$ as all the words in W are. This finishes proving that if all the events B_1, B_2, B_3, B_4, B_5, B_6 and B_7 hold, then our algorithm manages to reconstruct a piece the way we want it to. This means that the reconstructed piece is contained in the restriction of ξ to $[-4n, 4n]$, but contains the restriction of ξ to $[-n, n]$. In other words, the event $\mathcal{A}lg$ holds. □

Note that according to Lemma 2.1, the reconstruction algorithm works correctly as soon as all the events $B_1, B_2, \dots, B_7$ hold. Thus the probability that the algorithm does not work is bounded from above by the sum:

$$P(B_1^c) + P(B_2^c) + P(B_3^c) + P(B_4^c) + P(B_5^c) + P(B_6^c) + P(B_7^c). \tag{2.3}$$

Lemmas 2.2–2.11 provide upper bounds for $P(B_1^c)$–$P(B_7^c)$, respectively. As none is larger than a negative power in n, it follows that (2.3) can also be bounded by a negative power in n.

Lemma 2.2. *We have*

$$P(B_1^c) \le c_1 e^{-c_2 n},$$

where c_1 and c_2 are positive constants not depending on n.

Proof. Let $D_z = d(R_z, v_0)$ be the distance between v_0 and the vertex corresponding to R_z. Then, $\{D_z\}_{z\ge 0}$ is a simple random walk reflected at the origin, for which $P(D_z - D_{z-1} = 1 \mid D_{z-1} \neq$

$0) = 2/3$ and $P(D_z - D_{z-1} = -1 \mid D_{z-1} \neq 0) = 1/3$. Hence, $\{D_z\}_{z\geq 0}$ is a random walk with positive drift reflected at the origin. We can say

$$B_1 = \left[\cap_{z>4n}\{D_z > n\}\right] \cap \left[\cap_{s<-4n}\{D_{|s|} > n\}\right], \text{ so}$$
$$P(B_1^c) \leq 2\sum_{i>4n} P(D_i \leq n).$$

Let $\{T_z\}_{z\geq 0}$ be a random walk with the same transition probabilities as $\{D_z\}_{z\geq 0}$, starting at the origin. Then

$$P(D_z \leq n) \leq P(T_z \leq n),$$

and

$$P(B_1^c) \leq 2\sum_{i>4n} P(T_i \leq n).$$

Considering $T_i = X_1 + \cdots + X_i$, where $X_1, \ldots, X_i$ are i.i.d. random variables with $P(X_1 = 1) = 1 - P(X_1 = -1) = \frac{2}{3}$, we have

$$\begin{aligned} P(B_1^c) &\leq 2\sum_{i>4n} P(X_1 + \cdots + X_i \leq n) \\ &= 2\sum_{i>4n} P\left(\left(\frac{1}{4} - X_1\right) + \cdots + \left(\frac{1}{4} - X_i\right) \geq \frac{i-4n}{4}\right) \\ &\leq 2\sum_{i>4n} P\left(\left(\frac{1}{4} - X_1\right) + \cdots + \left(\frac{1}{4} - X_i\right) \geq 0\right). \end{aligned}$$

Let Y_i be equal to

$$Y_i := \frac{1}{4} - X_i.$$

Then the right side of the last inequality above is equal to

$$\sum_{i>4n}^{\infty} P(Y_1 + Y_2 + \cdots + Y_i \geq 0). \tag{2.4}$$

Note that $E[Y_i] = 0.25 - 0.\bar{3} = -0.08\bar{3}$, hence, by Large Deviation Theory, expression (2.4) must be exponentially small in n. Let us check out the details.

Recall that $P(Z \geq 0) \leq E[e^{Zt}]$ for any $t \geq 0$. Taking Z equal to $Y_1 + Y_2 + \cdots + Y_n$, we obtain

$$P(Y_1 + \cdots + Y_n \geq 0) \leq E[e^{Y_1 t}]^n, \tag{2.5}$$

for any $t \geq 0$.

Let $f(t)$ be the function $f(t) = E[e^{Y_1 t}]$, then if:

1. there is an open interval I around 0 such that $E[e^{Y_1 t}]$ is finite for all $t \in I$, and
2. the expectation of Y_1 is negative, i.e., $E[Y_1] < 0$,

there exists a small $t \geq 0$ such that $E[e^{Y_1 t}] \leq 1$.

The best possible exponential upper bound for (2.5) is the positive value for t which minimizes $E[e^{Y_1 t}]$.

For our definition of Y_1, i.e. $\left(Y_1 = \frac{1}{4} - X_1\right)$, the above two conditions hold, and $E[e^{Y_1 t}]$ reaches the minimum value for $t_0 = 0.091$, and $E[e^{Y_1 t_0}] = 0.99618$. It follows that

$$\begin{aligned} P(B_1^c) &\leq 2 \sum_{i=4n}^{\infty} (0.99618)^i \\ &= 2 \frac{(0.99618)^{4n}}{(1 - 0.99618)} \\ &= c_1 e^{-c_2 n}, \end{aligned}$$

where $c_1 = 523.56$ and $c_2 = 0.0153$. □

Lemma 2.3. *We have*

$$P(B_2^c) \leq 128 n^{(2-0.5k \log 3)}.$$

Proof. Let w_z denote the word:

$$w_z := \xi_z \xi_{z+1} \xi_{z+2} \dots \xi_{z+k \log n}$$

and $\bar{w}_z$ be the word

$$\bar{w}_z := \xi_z \xi_{z-1} \xi_{z-2} \dots \xi_{z-k \log n}.$$

Let B_{2,z_1,z_2} be the event where w_{z_1} is not equal to w_{z_2}, and $\bar{B}_{2,z_1,z_2}$ be the event where w_{z_1} is not equal to $\bar{w}_{z_2}$. Clearly:

$$B_2 = \left(\cap_{z_1 \neq z_2} B_{2,z_1,z_2}\right) \cap \left(\cap_{z_1,z_2} \bar{B}_{2,z_1,z_2}\right)$$

where the intersections above are taken over z_1, z_2 in $[-4n, 4n]$. This leads to

$$P(B_2^c) \leq \left(\sum_{z_1 \neq z_2} P(B_{2,z_1,z_2}^c) \right) + \left(\sum_{z_1,z_2} P(\bar{B}_{2,z_1,z_2}^c) \right) \tag{2.6}$$

where the sums above are taken with z_1, z_2 ranging over $[-4n, 4n]$, with n as an even number. For $z_1 \neq z_2$, we can always find at least $k \log n/2$ letters which are "all independent of each other in w_{z_1} and w_{z_2}". More precisely, since $z_1 \neq z_2$, there exists an integer subset $I \subset [0, k \log n]$ with at least $k \log n/2$ elements, so that

$$(z_1 + I) \cap (z_2 + I) = \emptyset.$$

Hence, using the fact that the scenery ξ is i.i.d. with 3 equiprobable colors, we find that if $z_1 \neq z_2$, then

$$P(B_{2,z_1,z_2}^c) = P(w_{z_1} = w_{z_2}) \leq \left(\frac{1}{3}\right)^{k \log n/2}. \tag{2.7}$$

A similar argument yields:

$$P(\bar{B}_{2,z_1,z_2}^c) = P(w_{z_1} = \bar{w}_{z_2}) \leq \left(\frac{1}{3}\right)^{k \log n/2}. \tag{2.8}$$

Applying inequalities (2.7) and (2.8) to inequality (2.6) yields:

$$P(B_2^c) \leq 128 n^2 \left(\frac{1}{3}\right)^{k \log n/2} = 128 n^{2-0.5k \log 3}.$$

The bound on the right side of the above inequality is a negative power of n as $2 - 0.5k \log 3$ is strictly negative. Hence, we just have to take $k > 0$ strictly larger than $4/\log 3$ to have a negative-power-in-n upper bound for $P(B_2^c)$. □

Lemma 2.4. *We have*

$$P(B_3{}^c) \leq c_1 n^{1-c_2 k_1},$$

where c_1 is a positive constant not depending on n.

Proof. Let B_{3z} be the event that $\{R(z - k_1 \log n) \notin R([z, +\infty))\}$. Then

$$B_3 = \cap_{z=-4n}^{4n} B_{3z}. \tag{2.9}$$

Note that the probability of B_{3z} does not depend on z. So Eq. (2.9) implies:

$$P(B_3{}^c) \leq \sum_{z=-4n}^{4n} P(B_{3z}^c) \leq 9n P(B_{3z}^c). \tag{2.10}$$

Thus taking z equal to $k_1 \log n$, we obtain

$$P(B_{3z}) = P(R(0)) \notin R([k_1 \log n, +\infty)) = P(v_0) \notin R([k_1 \log n, +\infty)).$$

As in the proof of Lemma 2.2, let the distance between $R(z)$ and v_0 be denoted by D_z so that

$$D_z := d(v_0, R(z)).$$

Again, as in 2.2, D_z is a simple random walk on $\mathbb{N}$ reflected at the origin. Let $\{T_z\}_{z \geq 0}$ be a random walk with the same transition probabilities as $\{D_z\}_{z \geq 0}$ and starting at the origin. We have

$$P(B_{3z}^c) \leq \sum_{z \geq k_1 \log n} P(T_z \leq 0). \tag{2.11}$$

Let $X_i := T_i - T_{i-1}$. Hence, we can use large deviations to bound

$$P(T_z \leq 0) = P(X_1 + X_2 + \cdots + X_z \leq 0).$$

With the same argument as in (2.2), we find

$$P(T_z \leq 0) \leq E(e^{-X_1 t})^z,$$

for any $t \geq 0$, where X_1 is a random variable such that

$$P(X_1 = 1) = 1 - P(X_1 = -1) = 2/3,$$

then we have $E[X_i] = +1/3$.

Minimizing $t \to E(e^{-X_1 t})$ with respect to t, we get

$$\min_{t \geq 0} E[e^{-X_1 t}] = 0.94281.$$

From (2.11), we obtain

$$P(B_{3z}^c) = \sum_{z \geq k_1 \log n} P(T_z \leq 0) \leq \sum_{z \geq k_1 \log n} 0.94281^z.$$

The last inequality above with inequality (2.10) together imply that

$$\begin{aligned} P(B_3^c) &\leq \frac{9n(0.94281)^{k_1 \log n}}{1 - 0.94281} \\ &= c_1 n e^{-c_2 k_1 \log n}, \quad \text{or} \\ &= c_1 n^{1-c_2 k_1}, \end{aligned}$$

where $c_1 = 157.37$ and $c_2 = 0.0589$. If we take the constant k_1 large such that $k_1 > \frac{1}{c_2}$, then our bound

$$P(B_3^c) \leq c_1 n^{1-c_2 k_1},$$

has a negative power in n and hence goes to 0 as n goes to infinity. □

Lemma 2.5. *We have*

$$P({B_4}^c) \leq c_1 e^{\log n (1-c_2 k_1)},$$

where $c_1 > 0$ is a positive constant not depending on n.

Proof. By symmetry, follows the same steps as for the proof of Lemma 2.4. □

Lemma 2.6. *We have B_5, which holds w.h.p:*

$$P(B_5^c) \leq \frac{c_5}{n}$$

where $c_5 > 0$ is a constant not depending on n.

Proof. Let $k_3 > 0$ be the constant

$$k_3 := k + 2k_1.$$

Let B_{51} be the event that the random walk S visits the points $-4n$ and $4n$ before time n^6, and B_{52z} be the event that the random walk S visits the point z at least n^{3k_3} times within n^{9k_3+9} time unit from the first visit to z. More precisely, let τ_{zi} be the ith visit by S to the point z. Hence

$$\tau_{z1} := \min\{t \mid S_t = z\}$$

and by induction on i:

$$\tau_{z(i+1)} := \min\{t > \tau_{zi} \mid S_t = z\}.$$

The event B_{52z} can now be described as the event of which difference

$$\tau_{zj} - \tau_{z1}$$

is less than or equal to n^{9k_3+9} for all $j \leq n^{3k_3}$. Let B_{53z} be the event such that the first n^{3k_3} visits of S to z have at least one straight path of length $k_3 \log n + 1$.

We have the following inclusion

$$B_{51} \cap \left(\cap_{z\in[-4n,4n]} B_{52z}\right) \cap \left(\cap_{z\in[-4n,4n]} B_{53z}\right) \subset B_5. \tag{2.12}$$

The above inclusion can be explained as follows. For any $z \in [-4n, 4n]$, from the event B_{51} the first visit to z by S takes place before time n^6. Then by B_{52z}, we get n^{3k_3} visits to z before an additional time n^{9k_3+9}. Hence, before time

$$\mathcal{T}^n := n^6 + n^{9k_3+9} \tag{2.13}$$

we have n^{3k_3} visits to z. According to the event B_{53z}, during those n^{3k_3} visits of S to z, there is at least one visit followed directly by a straight crossing of length $k_3 \log n + 1$. These crossings take place before time given in (2.13). In other words, we have just shown that when B_{51}, B_{52z} and B_{53z} all hold, then before time (2.13) there is a straight crossing by the random walk S of the interval $[z, z + k_3 \log n + 1]$. When there is such a straight crossing for each $z \in [-4n, 4n]$, then the event B_5 holds. This proves the inclusion (2.12). From inclusion (2.12), we obtain

$$P(B_5^c) \leq P(B_{51}^c) + \sum_{z \in [-4n,4n]} P(B_{52}^c) + \sum_{z \in [-4n,4n]} P(B_{53z}^c). \tag{2.14}$$

We can now apply the probability-bounds found in the next three lemmas to inequality (2.14) to find

$$P(B_5^c) \leq \frac{c_{51}}{n} + 8n\frac{c_{52}}{n^3} + 8ne^{-0.25n^{k_3}}.$$

In the sum, on the right side of last inequality above, the term with largest order is c_{51}/n. It follows, that there exists a constant $c_5 > 0$ not depending on n such that for all $n \in \mathbb{N}$, we have

$$P(B_5^c) \leq \frac{c_5}{n}. \quad \square$$

Lemma 2.7. *We have*

$$P(B_{51}^c) \leq \frac{c_{51}}{n}$$

where $c_{51} > 0$ is a constant not depending on n.

Proof. Let ν_i designate the first visit of the random walk S to the point i, i.e.,

$$\nu_i := \min\{t \mid S_t = i\},$$

and $\tau_i := \nu_i - \nu_{i-1}$. By the strong Markov property of the random walk, the sequence

$$\tau_1, \tau_2, \tau_3, \ldots$$

is a sequence of i.i.d. random variables. The random walk reaches the point $4n$ before time n^6 if and only if we have

$$\tau_{4n} \leq n^6.$$

This and a symmetric argument for $-4n$ leads to

$$P(B_{51}^c) \leq 2P(\tau_1 + \tau_2 + \cdots + \tau_{4n} > n^6)$$

and hence,

$$P(B_{51}^c) \leq 2P\left((\tau_1 + \tau_2 + \cdots + \tau_{4n})^{1/3} > n^2\right). \tag{2.15}$$

For positive numbers, the third power of the sum is always more than the sum of the third powers. Hence,

$$\tau_1 + \cdots + \tau_{4n} \leq (\tau_1^{1/3} + \cdots + \tau_{4n}^{1/3})^3$$

from which it follows that

$$(\tau_1 + \cdots + \tau_{4n})^{1/3} \leq \tau_1^{1/3} + \cdots + \tau_{4n}^{1/3}.$$

Applying the last inequality above to (2.15), we obtain

$$P(B_{51}^c) \leq 2P(\tau_1^{1/3} + \cdots + \tau_{4n}^{1/3} > n^2),$$

using Markov inequality yields

$$P(B_{51}^c) \leq 8\frac{E[\tau_1^{1/3}]}{n}.$$

The above bound is useful because the (1/3)th moment of τ_i is known to be finite. □

Lemma 2.8. *We have*

$$P(B_{52z}^c) \leq \frac{c_{52}}{n^3}$$

where $c_{52} > 0$ is a constant not depending on n or z.

Proof. Note that $P(B_{52_z}^c)$ does not depend on z. Hence, we can find a bound for $P(B_{520}^c)$ and this bound will be valid for all $P(B_{52_z}^c)$. Let T_i be the ith visit to the origin by S. We have

$$P(B_{520}^c) = P(T_1 + T_2 + \cdots + T_{n^{3k_3}} > n^{9k_3+9}).$$

Now, the expression on the right side of the equality above is equal to

$$P((T_1 + T_2 + \cdots + T_{n^{3k_3}})^{1/3} > n^{3k_3+3}). \tag{2.16}$$

Note that for non-negative terms, the third power of the sum is always larger than the sum of the third powers. Hence, in our case, taking the terms $T_i^{1/3}$ we get

$$T_1 + T_2 + \cdots + T_{n^{3k_3}} \leq \left(T_1^{1/3} + T_2^{1/3} + \cdots + T_{n^{3k_3}}^{1/3}\right)^3.$$

Taking the third root of the last inequality above we obtain

$$\left(T_1 + T_2 + \cdots + T_{n^{3k_3}}\right)^{1/3} \leq T_1^{1/3} + T_2^{1/3} + \cdots + T_{n^{3k_3}}^{1/3}. \tag{2.17}$$

Because of inequality (2.17), we find that the probability in expression (2.16) is less than or equal to

$$P(T_1^{1/3} + T_2^{1/3} + \cdots + T_{n^{3k_3}}^{1/3} > n^{3k_3+3}).$$

By the Markov inequality, we obtain

$$P(T_1^{1/3} + T_2^{1/3} + \cdots + T_{n^{3k_3}}^{1/3} > n^{3k_3+3}) \leq \frac{E[T_1^{1/3}]}{n^3}$$

and hence

$$P(B_{52z}^c) \leq \frac{E[T_1^{1/3}]}{n^3}.$$

The bound on the last inequality above is useful because $E[T_1^{1/3}]$ is known to be a finite number. □

Lemma 2.9. *We have*

$$P(B_{53z}^c) \leq e^{-0.25n^{k_3}}.$$

Proof. Let Y_i be the Bernoulli variable which is equal to one if and only if we have a straight crossing of length $k_3 \log n + 1$ right after the stopping time τ_{zj}, where $j = i(k_3 \log n + 1)$. Since, we take the stopping times $\tau_{z.}$ apart by at least $k_3 \log n + 1$, $Y_1, Y_2, \ldots$ are i.i.d. Also, the probability of a straight crossing is

$$P(Y_i = 1) = \left(\frac{1}{2}\right)^{k_3 \log n + 1} = \frac{1}{2n^{k_3}}.$$

The event B_{53_z} holds, as soon as at least one of the Y_i's is equal to 1 for $i = 1, 2, \ldots, n^{2k_3}$. Hence,

$$P(B^c_{53_z}) \leq P\left(\sum_{i=1}^{n^{2k_3}} Y_i = 0\right) = (1-q)^{n^{2k_3}}, \tag{2.18}$$

where

$$q = \frac{1}{2n^{k_3}}.$$

Note that

$$\left(1 - \frac{1}{2n^{k_3}}\right)^{2n^{k_3}}$$

converges to e^{-1} as $n \to \infty$. Applying this to (2.18), yields for n goes to infinity

$$P(B^c_{53_z}) \leq e^{-0.25n^{k_3}}. \quad \square$$

Lemma 2.10. *We have*

$$P({B_6}^c) \leq c_1 e^{-c_2 n},$$

where c_1 and c_2 are positive constants not depending on n.

Proof. Let $D_z = d(R_z, v_0)$ be the same as in the proof of Lemma 2.2. Recall that D_z is a simple random walk reflected at the origin with bias $+1/3$. We can write B_6 as

$$B_6 = \{\cap_{z>n/2} D_z > 0\} \cap \{\cap_{s<-n/2} D_s > 0\}.$$

Hence,

$$P({B_6}^c) \leq 2 \sum_{i=n/2}^{\infty} P(D_i = 0).$$

Let $\{T_z\}_{z\geq 0}$ be a random walk with the same transition probabilities as $\{D_z\}_{z\geq 0}$, and $X_1, \ldots, X_i$ be i.i.d. random variables with $P(X_1 = 1) = 1 - P(X_1 = -1) = 2/3$. Then once again

$$\begin{aligned} P({B_6}^c) &< 2 \sum_{i=n/2}^{\infty} P(T_i \leq 0) \\ &= 2 \sum_{i=n/2}^{\infty} P(X_1 + \cdots + X_i \leq 0) \end{aligned}$$

$$
\begin{aligned}
&= 2\sum_{i=n/2}^{\infty} P(-X_1 - \cdots - X_i \geq 0)\\
&\leq 2\sum_{i=n/2} (0.94281)^i,\\
&< 2\frac{2(0.94281)^n}{(1-0.94281)}\\
&= c_1 e^{-c_2 n},
\end{aligned}
$$

where $c_1 = 34.971$ and $c_2 = 0.0589$. □

Lemma 2.11. *We have*

$$P(B_7{}^c) \leq c_1 e^{-c_2 n},$$

where c_1 and c_2 are positive constants not depending on n.

Proof. Lemma 2.11 is a direct consequence of Lemmas 2.2 and 2.4. □

3. Final remarks and open problems

We have proposed a novel practical algorithm for reconstructing a piece of scenery from a given sequence of observations. Our algorithm is based on one of the techniques for DNA reconstruction, which is known as "polymerase chain reaction" (PCR). For retrieving a finite piece of a 3-color scenery of length n, small pieces of length order $\ln n$ are first reconstructed and then assembled together to form the required piece. The algorithm takes a polynomial amount of time for reconstructing a 3-color scenery.

The open problems to this research include the fundamental properties of the distribution of the scenery for a plausible reconstruction, which are still not defined precisely, for example, if the entropy grows linearly with the size of the scenery, a universal reconstruction algorithm exists for solving the scenery reconstruction problem?

References

[1] J.M. Bartlett, D. Stirling, A short history of the polymerase chain reaction, Methods in Molecular Biology 226 (2003) 3–6.

[2] A. Hart, H. Matzinger, Markers for error-corrupted observations, Stochastic Processes and their Applications 116 (2006) 807–829.

[3] J. Lember, H. Matzinger, Information recovery from a randomly mixed up message-text, Electronic Journal of Probability 13 (2008) 396–466.

[4] J. Lember, H. Matzinger, (2003) Reconstructing a 2-color scenery by observing it along a recurrent random walk path with bounded jumps. Eurandom (in preparation).

[5] Lindenstrauss, Indistinguishable sceneries, Random Structures Algorithms 14 (1999) 71–86.

[6] M. Löwe, H. Matzinger, Scenery reconstruction in two dimensions with many colors, The Annals of Applied Probability 12 (4) (2002) 1322–1347.

[7] M. Löwe, H. Matzinger, F. Merkl, Reconstructing a multicolor random scenery seen along a random walk path with bounded jumps, Electronic Journal of Probability 15 (2004) 436–507.

[8] H. Matzinger, Reconstructing a three-color scenery by observing it along a simple random walk path, Random Structures Algorithms 15 (2) (1999) 196–207.

[9] H. Matzinger, Reconstructing a 2-color scenery by observing it along a simple random walk path, The Annals of Applied Probability 15 (2005) 778–819.

[10] H. Matzinger, S.W.W. Rolles, Reconstructing a random scenery observed with random errors along a random walk path, Probability Theory and Related Fields 125 (4) (2003) 539–577.

[11] H. Matzinger, S.W.W. Rolles, Retrieving random media, Probability Theory and Related Fields 136 (2006) 469–507.

[12] H. Matzinger, S.W.W. Rolles, Finding blocks and other patterns in a random coloring of ℤ, Random Structures Algorithms 28 (2006) 37–75.

[13] H. Matzinger, S.W.W. Rolles, Reconstructing a piece of scenery with polynomially many observations, Stochastic Processes and their Applications 107 (2) (2003) 289–300.

[14] S. Popov, H. Matzinger, Detecting a local perturbation in a continuous scenery, Electronic Journal of Probability 12 (2008) 1103–1120.