

ASYMPTOTIC HITTING TIME FOR A SIMPLE EVOLUTIONARY MODEL OF PROTEIN FOLDING

VÉRONIQUE LADRET,* *Université Claude Bernard Lyon 1*

Abstract

We consider two versions of a simple evolutionary algorithm (EA) model for protein folding at zero temperature, namely the $(1 + 1)$ -EA on the LeadingOnes problem. In this schematic model, the structure of the protein, which is encoded as a bit-string of length n , is evolved to its native conformation through a stochastic pathway of sequential contact bindings. We study the asymptotic behavior of the hitting time, in the mean case scenario, under two different mutations: the *one-flip*, which flips a unique bit chosen uniformly at random in the bit-string, and the *Bernoulli-flip*, which flips each bit in the bit-string independently with probability c/n , for some $c \in \mathbb{R}^+$ ($0 \leq c \leq n$). For each algorithm, we prove a law of large numbers, a central limit theorem, and compare the performance of the two models.

Keywords: Evolutionary algorithm; Markov chain; protein folding

2000 Mathematics Subject Classification: Primary 60J10; 60F05; 92D20; 92C05

1. Introduction

Evolutionary algorithms (EAs) are adaptive heuristic search algorithms. They are based on the mechanisms of natural selection and are widely used in a great variety of problems, for example population genetics, machine learning, and optimization. The task of the EA is to search a *fitness landscape* for maximal values. A population of individuals, considered as candidate solutions to the given problem, is evolved through steps of variation and steps of selection. Each individual receives a numerical evaluation, called its *fitness score*. The dynamics of the EA simulate, supposedly as in natural systems, the survival of the fittest among the individuals. Thus, individuals of maximum fitness (in this sense) are sought.

Despite their numerous heuristic successes, mathematical results describing the behavior of EAs are rather sparse. Among the exceptions are [4], [5], [7]–[9], [16], [18], and [19].

Since EAs usually exhibit complicated dynamics, complexity results are difficult to reach and a common approach is to consider simplified cases. Among the simplified EAs are the so-called $(1 + 1)$ -EAs. These were studied in [2], [11]–[14], [17], and [19]. In this paper, we study the time of convergence of two versions of a specific $(1 + 1)$ -EA, namely the $(1 + 1)$ -EA on the LeadingOnes problem. One of the main motivations for studying these algorithms is that they can be used as simple models for the protein-folding problem. Indeed, the $(1 + 1)$ -EAs we focus on apply directly to the model of protein-structure prediction at zero temperature that has been proposed by the biophysicists Bakk *et al.* [3].

Received 29 January 2003; revision received 16 July 2004.

* Current address: Département de mathématiques et de statistique, Université de Montréal, CP 6128 succursale Centre-ville, Montréal, Québec H3C 3J7, Canada. Email address: ladretve@dms.umontreal.ca

1.1. The physical model

Proteins typically fold to a unique native or biologically active conformation on time-scales from 10^{-3} s to 1 s. However, if the dynamics of the folding process were to follow a random search in the conformation space it would result in astronomical folding time-scales; this is known as Levinthal's paradox [15]. So, how do proteins fold to their native state? This is one of the intriguing problems of biophysics. Anfinsen [1] showed that the native state is genetically, as well as thermodynamically, determined, i.e. it corresponds to the conformation in which the Gibbs free energy of the whole system is lowest.

There are many hypotheses concerning the transition state. One of the views is that the transition-state dynamics consists of a stochastic pathway that carries the polypeptide (protein) to the native state through a guided descent within the Gibbs-free-energy landscape (see [10] and [20]).

The protein-like model proposed by Bakk *et al.* [3] can be described as follows. The polypeptide chain is equipped with n contact points c_1, \dots, c_n that we will call nodes. For i from 1 to n , c_i is assigned a binary contact variable ϕ_i that indicates whether it is folded ($\phi_i = 1$) or unfolded ($\phi_i = 0$). In consequence, the conformation of the protein is entirely determined by the bit-string of length n , $\phi = (\phi_1, \phi_2, \dots, \phi_n)$ (there is a bijective mapping from the conformation space onto $\{0, 1\}^n$), and the native state corresponds to the bit-string for which $\phi_i = 1$ for all i , $1 \leq i \leq n$, i.e. $(1, \dots, 1)$.

Let i_0 denote the smallest $i \in \{1, \dots, n\}$ for which c_i is unfolded, i.e. for which $\phi_i = 0$. We refer to the set of contact points $\{c_{i_0}, c_{i_0+1}, \dots, c_n\}$ as the open part of the protein. An assumption made about the dynamics of the folding process is that each individual node is assigned an energy of $-\varepsilon_0$ if $i < i_0$, and 0 otherwise. This can be implemented through the Hamiltonian

$$\mathcal{H} = -\varepsilon_0(\phi_1 + \phi_1\phi_2 + \dots + \phi_1 \dots \phi_n).$$

This Hamiltonian can be rewritten in terms of the LeadingOnes function L , defined on the space of conformations $\{0, 1\}^n$, which counts the length of the longest prefix of ones in the bit-string:

$$L(x) = \max\{k \geq 1 : \text{for all } i, 1 \leq i \leq k, x_i = 1\} \cup \{0\}.$$

Indeed,

$$\mathcal{H} = -\varepsilon_0 L(\phi) = -\varepsilon_0(i_0 - 1). \quad (1.1)$$

In this model, which is also known as the 'zipper-model', there is no energy associated with the open part of the protein. In fact, a descent through the energy landscape proceeds via the folding of the left-most incorrectly folded substructure, i.e. node c_{i_0} , while the correctly folded substructures preceding it, i.e. nodes c_i for $i < i_0$, remain unchanged. This means that the folding events occur in a specific order: they behave like the individual locks in a zipper. In the bit-string framework, lowering the Gibbs free energy is exactly equivalent to increasing the size of the longest prefix of ones.

The algorithm proposed by Bakk *et al.* [3] to search the state space $\{0, 1\}^n$ for the conformations of lowest energy, i.e. the native state, is based on the Monte Carlo Metropolis method [6]. Let T denote the temperature of the whole system, k the usual Boltzmann constant, and let $\beta = 1/kT$. The algorithm proceeds iteratively, as follows. The individual (i.e. the bit-string) at time k , X_k , which determines the protein conformation ϕ at time k , undergoes a mutation to a new conformation X'_k through a stochastic process that will be described later. Then X'_k is

selected to form the new individual X_{k+1} , at time $k + 1$, with probability

$$P_{\text{accept}} = \min(1, \exp(-\beta \Delta \mathcal{H})), \quad \text{where} \\ \Delta \mathcal{H} = \mathcal{H}(X'_k) - \mathcal{H}(X_k).$$

Otherwise, X_{k+1} is taken to be the same as the old configuration X_k .

In this paper, we concentrate on the Monte Carlo Metropolis model considered at zero temperature. This algorithm is directly related to the (1 + 1)-EA on the LeadingOnes problem. We recall that the dynamics of (1 + 1)-EAs can be formalized through discrete Markov chains as follows.

1.2. The (1 + 1)-EA approach to native conformation prediction

In the protein model, we wish to minimize the Hamiltonian \mathcal{H} , which is equivalent, according to (1.1), to maximizing the LeadingOnes function. More generally, the goal of (1 + 1)-EAs is to optimize some fitness function $f: \{0, 1\}^n \rightarrow \mathbb{R}$. The algorithm proceeds as follows: a unique individual, or bit-string, is evolved according to the following two-step iterative process.

Step 1. Mutation. As in the Monte Carlo Metropolis method, at every evolutionary step (or generation), the individual chosen from the population at time k , X_k , undergoes a random walk to a new individual X'_k .

Step 2. Selection. The fitness values of X_k and X'_k are evaluated. The individual with the highest fitness value $f(\cdot)$ is selected to form the generation X_{k+1} at time $k + 1$. That is,

$$\text{if } f(X'_k) > f(X_k) \text{ then } X_{k+1} = X'_k; \text{ otherwise } X_{k+1} = X_k. \quad (1.2)$$

The notation (1 + 1)-EA accounts for the fact that we select between one ‘parent’ and one ‘child’. Here, as in the protein-folding dynamics of Bakk *et al.* [3], we focus on the mean case scenario, in which the first individual X_0 is chosen uniformly, and at random, from $\{0, 1\}^n$. When the fitness function is precisely L , this algorithm will be denoted by $(1 + 1)_L$ -EA. We remark that, in the case of a fitness landscape with local maxima, the (1 + 1)-EA method could end in a suboptimal search. However, we will only consider the LeadingOnes problem whose fitness landscape has no local maxima, and neglect this possibility.

In the literature, there is no actual consensus, in the definition of (1 + 1)-EAs, on the selection rule. It is sometimes taken to be the following slightly different one.

Step 2*. Selection. The inequality in (1.2) is weakened. That is,

$$\text{if } f(X'_k) \geq f(X_k) \text{ then } X_{k+1} = X'_k; \text{ otherwise } X_{k+1} = X_k. \quad (1.3)$$

For example, Garnier *et al.* [13] considered the first version of the selection rule (1.2), whereas Droste *et al.* [11], [12] focused on the second version (1.3). In the LeadingOnes framework, in order to discriminate between the two versions we will denote by $(1 + 1)_L^*$ -EA the analog of $(1 + 1)_L$ -EA that uses selection rule (1.3). The only difference between these two very similar algorithms is that $(1 + 1)_L^*$ -EA accepts candidates X'_k whose energy (fitness) is the same as that of X_k , whereas $(1 + 1)_L$ -EA does not. We point out that this seemingly small difference may result in significant changes in the behavior of the (1 + 1)-EA: see, for example, [14], where Jansen and Wegener exhibit two fitness functions such that the first (1 + 1) version (with selection rule (1.2)) maximizes one function in polynomial time and takes exponential time to maximize the other, while the second version (using selection rule (1.3)) has the opposite behavior.

1.3. Statement of the results

Let T_n and \widehat{T}_n respectively denote the hitting times before some optimal conformation or individual (with respect to the fitness function) is sampled by the $(1+1)$ -EA and the $(1+1)_L^*$ -EA. We focus on both the $(1+1)_L^*$ -EA and the $(1+1)_L$ -EA in the mean case scenario, under two different kinds of mutation: the *one-flip*, which flips a unique bit chosen uniformly at random in the bit-string, and the *Bernoulli-flip*, which flips each bit in the bit-string independently with probability c/n .

As was briefly mentioned in the introduction, there has already been some work on the complexity of some $(1+1)$ -EAs. From [11], $E(\widehat{T}_n) = \Theta(n \ln n)$ for the Bernoulli-flip applied to a linear fitness function on $\{0, 1\}^n$ (i.e. for some positive constants a_1 and a_2 and some $n_0 \in \mathbb{N}$, $a_1 n \ln n \leq \widehat{T}_n \leq a_2 n \ln n$ for all $n \geq n_0$). From [12], the LeadingOnes function in the Bernoulli-flip scenario is solvable in mean time $\Theta(n^2)$, and there are constants $C_1, C_2 > 0$ such that the probability that \widehat{T}_n is outside the interval $[C_1 n^2, C_2 n^2]$ is exponentially small.

Garnier *et al.* [13] studied the OneMax function $|\cdot|$, which counts the number of ones in the bit-string in the one-flip and the Bernoulli-flip frameworks, i.e.

$$|x| = \sum_{i=1}^n x_i, \quad \text{with } x = (x_1, \dots, x_n) \in \{0, 1\}^n.$$

In the one-flip case, $(T_n - n \ln n)/n$ converges in distribution to $-\ln 2 - \ln Z$. In the Bernoulli-flip case, $(T_n - c^{-1} e^c n \ln n)/n$ converges in distribution to $-c^{-1} e^c \ln Z + C(c)$, where the law of Z is exponential with parameter 1 and $C(c)$ is some c -dependent constant.

Here, we prove the analog of the result of [13] for the LeadingOnes problem. This improves on the result of [12]. We prove a law of large numbers, a central limit theorem, and we compare the performance of the two models. Finally, we prove that the distribution of the hitting time of the $(1+1)_L^*$ -EA, \widehat{T}_n , is the same as that of T_n in both the one-flip and the Bernoulli-flip scenarios.

Theorem 1.1. (The one-flip case.)

- (i) For $n \geq 1$, $E(T_n) = \frac{1}{2}n^2$;
- (ii) As $n \rightarrow \infty$, $T_n/E(T_n)$ converges in probability to 1.
- (iii) As $n \rightarrow \infty$, $(T_n - E(T_n))/n^{3/2}$ converges in distribution to a centered Gaussian random variable of variance $\frac{3}{4}$.

Theorem 1.2. (The Bernoulli-flip case.)

- (i) As $n \rightarrow \infty$, $E(T_n) \sim m(c)n^2$, with

$$m(c) := \frac{e^c - 1}{2c^2}.$$

- (ii) As $n \rightarrow \infty$, $T_n/E(T_n)$ converges in probability to 1.
- (iii) Furthermore, $(T_n - m(c)n^2)/n^{3/2}$ converges in distribution to a centered Gaussian random variable of variance $\sigma^2(c)$, with

$$\sigma^2(c) := \frac{3(e^{2c} - 1)}{8c^3}.$$

Note that $m(c) > \frac{1}{2}$ for every $c > 0$.

Corollary 1.1. *As $n \rightarrow \infty$, $E(T_n)$ in the Bernoulli-flip case is greater than $E(T_n)$ in the one-flip case, for any value of c .*

Theorem 1.3. *In both the one-flip and the Bernoulli-flip cases, T_n and \widehat{T}_n have the same distribution.*

2. Proof of Theorem 1.1

The law of T_n , conditioned by $|X_0|$, is the law of a sum of geometric random variables – see Lemma 2.1, below. This yields part (i) of the theorem. Since the central limit theorem implies the law of large numbers, we then prove the central limit theorem of part (iii).

The distribution of T_n can be deduced from a simple observation. In the $(1 + 1)$ -EA on the LeadingOnes problem, a mutation is accepted if and only if it adds one to the number of leading 1s. As a consequence, in the one-flip framework, the Markov chain $(X_k)_{k \geq 0}$ jumps (i.e. accepts a transition to a new conformation) when the left-most 0 is flipped – the other flips leave the chain unchanged. Thus, the 0s in X_0 are successively flipped, from left to right, until the algorithm hits the optimal individual $(1, 1, \dots, 1)$.

Henceforth, $\varepsilon = 1/n$ when we deal with algorithms on strings of length n ; $|x|$ is the number of 1s in $x = (x_1, \dots, x_n) \in \{0, 1\}^n$; the geometric law $\mathcal{G}(p)$ of parameter p is defined by

$$\mathcal{G}(p) = \sum_{n \geq 1} p(1 - p)^{n-1} \delta_n,$$

where δ_n is the Kronecker delta; and the negative binomial law of parameter (k_0, p) , $\mathcal{NB}(k_0, p)$, puts the following mass on $k \geq k_0$:

$$\binom{k - 1}{k - k_0} p^{k_0} (1 - p)^{k - k_0}.$$

Lemma 2.1. *If $|X_0| = n - k_0$ then T_n is the sum of k_0 independent, identically distributed $\mathcal{G}(\varepsilon)$ -random variables. Thus, the law of T_n is negative binomial of parameter (k_0, ε) .*

Proof. Let $\tau_0 = 0$ and, for every $k \geq 0$, let

$$\tau_{k+1} = \inf\{i \geq \tau_k : X_i \neq X_{\tau_k}\}, \quad \sigma_{k+1} = \tau_{k+1} - \tau_k, \quad \widetilde{X}_k = X_{\tau_k}.$$

In words, \widetilde{X}_k denotes the position of the chain after its k th jump, and $|\widetilde{X}_k| = n - k_0 + k$. The left-most 0 of \widetilde{X}_k is flipped once the chain’s position has remained \widetilde{X}_k for a time σ_{k+1} . Thus, $(\sigma_k)_k$ is independent and identically distributed, and has law $\mathcal{G}(\varepsilon)$. It remains to note that

$$T_n = \sigma_1 + \dots + \sigma_{k_0},$$

which completes the proof.

Proof of Theorem 1.1(iii). Let E_k denote expectation under conditioning on $\{|X_0| = n - k\}$. Since X_0 is uniform, the law μ of $|X_0|$ is binomial of parameter $(n, \frac{1}{2})$. From Lemma 2.1, under P_k , T_n is the sum of k independent and identically distributed geometric random variables of parameter ε . Thus,

$$E_k(e^{-\alpha T_n}) = e^{-\alpha k} \varepsilon^k [1 - (1 - \varepsilon)e^{-\alpha}]^{-k}. \tag{2.1}$$

Let $\Theta_n = (T_n - n^2/2)/n^{3/2}$ and compute the decomposition of its Laplace transform along the values of $|X_0|$. Recall that, here, we consider the mean-case scenario, in which the initial

conformation X_0 is chosen uniformly at random in $\{0, 1\}^n$; thus, we compute the Laplace transform by using (2.1) and by conditioning on the different values taken by $|X_0|$, i.e. the events $\{|X_0| = n - k\}$, $k = 0, \dots, n$. This decomposition, the explicit form of μ , and (2.1) together yield

$$E(e^{-\alpha\Theta_n}) = e^{\sqrt{n}\alpha/2} 2^{-n} \sum_{k=0}^n \binom{n}{k} e^{-\alpha k/n^{3/2}} \varepsilon^k [1 - (1 - \varepsilon) e^{-\alpha/n^{3/2}}]^{-k}.$$

This can be rewritten as

$$E(e^{-\alpha\Theta_n}) = e^{\sqrt{n}\alpha/2} 2^{-n} (1 + \beta_n)^n,$$

with

$$\beta_n = \varepsilon e^{-\alpha/n^{3/2}} [1 - (1 - \varepsilon) e^{-\alpha/n^{3/2}}]^{-1}.$$

Recall that $\varepsilon = 1/n$. The expansion of β_n reads

$$\beta_n = 1 - \frac{\alpha}{\sqrt{n}} + \frac{\alpha^2}{n} + o\left(\frac{1}{n}\right),$$

which implies that

$$E(e^{-\alpha\Theta_n}) \rightarrow e^{3\alpha^2/8}$$

as $n \rightarrow \infty$. This concludes the proof.

3. Proof of Theorem 1.2

We first describe the law of T_n conditioned on the values taken by L along the path of $(X_k)_k$ up to time T_n , that is, until the optimal individual $(1, 1, \dots, 1)$ is hit. This law is the law of a sum of independent geometric random variables – see Lemma 3.3, below. We deduce the overall law of T_n and state it in Proposition 3.1, yielding part (i) of Theorem 1.2. As in the previous section, since the central limit theorem implies the law of large numbers, we then prove the central limit theorem of part (iii).

We recall that the $(1 + 1)$ -EA, in the LeadingOnes framework, accepts a mutation if and only if the number of leading 1s is increased. Hence, the dynamics of the Bernoulli-flip algorithm proceeds as follows: the chain jumps to a new individual at time $k + 1$ if and only if the leading 1s of X_k are left unchanged and its left-most 0 is flipped, no matter which values are taken by the other bits.

Henceforth, $\varepsilon = c/n$ when we deal with algorithms on strings of length n . For all $i \geq 0$, let $p(n, i) = \varepsilon(1 - \varepsilon)^i$. As in the one-flip framework, \tilde{X}_k denotes the position of the chain after its k th jump. We also keep the same definitions for σ_k and τ_k .

For all $k \geq 0$, let

$$\ell_k = L(\tilde{X}_k),$$

and let 1^{ℓ_0} and Y_0 be bit-strings (1^{ℓ_0} a string of ℓ_0 1s) such that

$$X_0 = (1^{\ell_0}, 0, Y_0).$$

For $k \geq 1$, define further bit strings Y_k and W_k in such a way that

$$\tilde{X}_k = (1^{\ell_{k-1}}, 1, W_k) = (1^{\ell_k}, 0, Y_k).$$

Lemmas 3.1 and 3.2 are needed in Lemma 3.3 to compute the law of T_n conditioned on (ℓ_j) .

Lemma 3.1. (i) For all $k \geq 0$, σ_k depends on the past only through the last score ℓ_{k-1} . That is, the law of σ_k conditioned on $(X_t)_{t < \tau_k}$ is the law of σ_k conditioned on ℓ_{k-1} .

(ii) For all $k \geq 0$, given that $\{\ell_{k-1} = i\}$ occurs, the law of σ_k is $\mathcal{G}(p(n, i))$.

Proof. The sojourn time σ_k is the time the algorithm takes to jump from \tilde{X}_{k-1} to \tilde{X}_k . The left-most 0 of \tilde{X}_{k-1} is in position $\ell_{k-1} + 1$. Thus, we need to flip the $(\ell_{k-1} + 1)$ th bit while leaving the first ℓ_{k-1} bits unchanged. Hence,

$$P(\sigma_k = t \mid \tilde{X}_0, \dots, \tilde{X}_{k-1}) = \varepsilon(1 - \varepsilon)^{\ell_{k-1}}[1 - \varepsilon(1 - \varepsilon)^{\ell_{k-1}}]^{(t-1)}.$$

Lemma 3.2. For all $k \geq 1$, let $\hat{\mathcal{F}}_k = \sigma\{\sigma_i, \ell_j : i \leq k, j \leq k - 1\}$. Then, the law of ℓ_k conditioned on $\hat{\mathcal{F}}_k$ is the law of ℓ_k conditioned on ℓ_{k-1} .

Proof. Let P_i denote probability, conditioned on the event $\{\ell_0 = i\}$. Using the strong Markov property,

$$P(\ell_k = i_k \mid \sigma_k = t_k, \ell_{k-1} = i_{k-1}, \dots, \sigma_1 = i_1, \ell_0 = i_0) = P(\ell_1 = i_k \mid \sigma_1 = t_k, \ell_0 = i_{k-1}). \tag{3.1}$$

Since $\{\ell_1 = i_1, \sigma_1 = t, \ell_0 = i_0\} = \{L(X_t) = i_1, L(X_{t-1}) = \dots = L(X_0) = i_0\}$ and $\{\sigma_1 = t, \ell_0 = i_0\} = \{L(X_t) \neq i_0, L(X_{t-1}) = \dots = L(X_0) = i_0\}$, by using the Markov property on $(L(X_t)_t)$ we derive the following expression:

$$P(\ell_1 = i_k \mid \sigma_1 = t_k, \ell_0 = i_{k-1}) = \frac{P(L(X_1) = i_k \mid L(X_0) = i_{k-1})}{P(L(X_1) \neq i_{k-1} \mid L(X_0) = i_{k-1})}.$$

This quantity is independent of t_k ; hence, if we reconsider (3.1), we find that

$$P(\ell_k = i_k \mid \sigma_k = t_k, \ell_{k-1} = i_{k-1}, \dots, \sigma_1 = i_1, \ell_0 = i_0) = P(\ell_k = i_k \mid \ell_{k-1} = i_{k-1}).$$

Lemma 3.3. Conditioned on $\{\ell_0 = i_0, \dots, \ell_{J-1} = i_{J-1}, \ell_J = n\}$, T_n is the sum of J independent geometric random variables with respective parameters $p(n, i_0), \dots, p(n, i_{J-1})$.

Proof. Given the successive LeadingOnes scores $\ell_0 = i_0, \dots, \ell_J = n$ until the optimal individual is hit, $T_n = \sum_{k=1}^J \sigma_k$. Thus,

$$P(T_n = t \mid \ell_J = n, \dots, \ell_0 = i_0) = \sum_{t_1 + \dots + t_k = t} P(\sigma_J = t_J, \dots, \sigma_1 = t_1 \mid \ell_J = n, \dots, \ell_0 = i_0). \tag{3.2}$$

Using Lemmas 3.2 and 3.1 we can derive the following, by induction:

$$\begin{aligned} P(\ell_J = n, \sigma_J = t_J, \dots, \sigma_1 = t_1, \ell_0 = i_0) \\ = \prod_{k=1}^J P(\ell_k = i_k \mid \ell_{k-1} = i_{k-1})P(\sigma_k = t_k \mid \ell_{k-1} = i_{k-1}). \end{aligned}$$

Hence, since $\prod_{k=1}^J P(\ell_k = i_k \mid \ell_{k-1} = i_{k-1}) = P(\ell_J = n, \dots, \ell_0 = i_0)$,

$$P(\sigma_1 = t_1, \dots, \sigma_J = t_J \mid \ell_J = n, \dots, \ell_0 = i_0) = \prod_{k=1}^J P(\sigma_k = t_k \mid \ell_{k-1} = i_{k-1}).$$

We can substitute this equation into (3.2), to obtain

$$P(T_n = t \mid \ell_J = n, \dots, \ell_0 = i_0) = \sum_{t_1 + \dots + t_k = t} \prod_{k=1}^J P(\sigma_k = t_k \mid \ell_{k-1} = i_{k-1}).$$

Let $q(n, i_k)$ denote the probability distribution of $\mathcal{G}(p(n, i_k))$. Then, as we recognize a product of convolution in the last equation, using Lemma 3.1 we can write

$$P(T_n = t \mid \ell_J = n, \dots, \ell_0 = i_0) = q(n, i_{J-1}) * \dots * q(n, i_0)(t).$$

Thus, given that the search jumps J times before the target $(1, 1, \dots, 1)$ is hit, and given that $\{\ell_0 = i_0, \dots, \ell_J = n\}$ occurs, T_n follows the same distribution as the sum of J independent random variables respectively distributed as $\mathcal{G}(p(n, i_0)), \dots, \mathcal{G}(p(n, i_{J-1}))$.

Now, let us focus on $P(\ell_0 = i_0, \dots, \ell_{J-1} = i_{J-1}, \ell_J = n)$. In order to compute this quantity, we need the following lemma.

Lemma 3.4. *Let $k \geq 1$. If X_0 is chosen uniformly in $\{0, 1\}^n$ then, given that $\{\ell_{k-1} = i\}$ occurs, W_k follows the uniform distribution on $\{0, 1\}^{n-i-1}$, i.e.*

$$\mathcal{L}(W_k \mid \ell_{k-1} = i) = \mathcal{U}(\{0, 1\}^{n-i-1}),$$

where $\mathcal{L}(W_k \mid \ell_{k-1} = i)$ denotes the law of W_k conditioned on the event $\{\ell_{k-1} = i\}$ and $\mathcal{U}(\{0, 1\}^{n-i-1})$ denotes the uniform distribution on $\{0, 1\}^{n-i-1}$.

Proof. Let us focus on the case in which $k = 1$ and let μ denote the probability distribution of Y_0 given $\{\ell_0 = i\}$. As X_0 is chosen uniformly in $\{0, 1\}^n$, μ is the uniform distribution on $\{0, 1\}^{n-i-1}$. Now,

$$P_i(W_1 = w) = \sum_{t \geq 1} P_i(\tilde{X}_1 = (1^i, 1, w), \sigma_1 = t) \tag{3.3}$$

and, since $\{X_{\sigma_1} = (1^i, 1, w), \sigma_1 = t, L(X_0) = i\} = \{X_t = (1^i, 1, w), L(X_{t-1}) = \dots = L(X_0) = i\}$, by using the Markov property, (3.3) can be rewritten as

$$P_i(W_1 = w) = \sum_{t \geq 1} P_i(X_1 = (1^i, 1, w)) P_i(L(X_1) = L(X_0))^{t-1}.$$

Hence,

$$P_i(W_1 = w) = \frac{P_i(X_1 = (1^i, 1, w))}{P_i(L(X_1) > L(X_0))}. \tag{3.4}$$

We recall that the Markov chain jumps from X_0 to a conformation of higher fitness at time 1 if none of the first ℓ_0 1s of X_0 are flipped while its left-most 0 is. Hence,

$$P_i(L(X_1) > L(X_0)) = \varepsilon(1 - \varepsilon)^i. \tag{3.5}$$

On the other hand, as $P_i(Y_0 = u) = \mu(u) = 1/2^{n-i-1}$,

$$P_i(X_1 = (1^i, 1, w)) = \frac{1}{2^{n-i-1}} \sum_{u \in \{0, 1\}^{n-i-1}} P(X_1 = (1^i, 1, w) \mid X_0 = (1^i, 0, u)). \tag{3.6}$$

If $d(w, u)$ denotes the Hamming distance between w and u , then we can write

$$P(X_1 = (1^i, 1, w) \mid X_0 = (1^i, 0, u)) = \varepsilon(1 - \varepsilon)^i \varepsilon^{d(w,u)} (1 - \varepsilon)^{n-i-1-d(w,u)}. \tag{3.7}$$

Finally, (3.4), (3.5), (3.6), and (3.7), together with

$$\sum_{u \in \{0,1\}^{n-i-1}} \varepsilon^{d(w,u)} (1 - \varepsilon)^{n-i-1-d(w,u)} = 1,$$

yield $P_i(W_1 = w) = 1/2^{n-i-1}$.

Now, using the strong Markov property, we can derive the proof for any $k \geq 2$.

Lemma 3.5. *If X_0 is chosen uniformly and at random in the state space, then, for all $k \geq 1$, the conditional distribution of ℓ_k , given $\{\ell_{k-1} = j_{k-1}\}$, satisfies*

$$P(\ell_k = j_k \mid \ell_{k-1} = j_{k-1}) = \begin{cases} 2^{-(j_k - j_{k-1})} & \text{if } j_{k-1} + 1 \leq j_k < n, \\ 2^{-(n - j_{k-1} - 1)} & \text{if } j_k = n. \end{cases}$$

Proof. This is a direct consequence of Lemma 3.4.

Now that we know the probability distribution of the sequence of successive LeadingOnes scores until the target individual is hit, as well as the distribution of T_n conditioned on the values taken by these LeadingOnes scores, we can compute the probability distribution of T_n as follows.

Proposition 3.1. *If X_0 is chosen uniformly at random in $\{0, 1\}^n$, then the probability distribution of T_n satisfies*

$$P(T_n = t) = \frac{1}{2^n} \sum_J \sum_{0 \leq i_0 \leq i_1 \leq \dots \leq i_J = n} q(n, i_{J-1}) * \dots * q(n, i_0)(t). \tag{3.8}$$

Proof. That X_0 is chosen uniformly and at random in the search space implies that $P(\ell_0 = i_0) = 1/2^{i_0+1}$. Thus, applying Lemma 3.5,

$$P(\ell_0 = i_0, \dots, \ell_J = n) = \frac{1}{2^n}.$$

The result is then a direct consequence of Lemma 3.3.

Proof of Theorem 1.2(iii). Set $\Theta_n = (T_n - n^2(e^c - 1)/2c^2)/n^{3/2}$. Then,

$$E(\exp(-\alpha \Theta_n)) = \exp\left(\alpha \frac{\sqrt{n}}{2c^2} (e^c - 1)\right) E\left(\exp\left(-\alpha \frac{T_n}{n^{3/2}}\right)\right) \tag{3.9}$$

and, according to the distribution of T_n given by (3.8),

$$E\left(\exp\left(-\alpha \frac{T_n}{n^{3/2}}\right)\right) = \frac{1}{2^n} \sum_J \sum_{i_0 < \dots < i_{J-1}} E\left(\exp\left(-\frac{\alpha}{n^{3/2}} [\mathcal{G}(p(n, i_0)) + \dots + \mathcal{G}(p(n, i_{J-1}))]\right)\right).$$

Since the variables $(\mathcal{G}(p(n, i_k)))_{i_k}$ are independent,

$$\begin{aligned} E\left(\exp\left(-\alpha \frac{T_n}{n^{3/2}}\right)\right) &= \frac{1}{2^n} \sum_J \sum_{i_0 < \dots < i_{J-1}} \prod_{k=0}^{J-1} E\left(-\frac{\alpha}{n^{3/2}} \mathcal{G}(p(n, i_k))\right) \\ &= \frac{1}{2^n} \prod_{k=0}^{n-1} \left(1 + \phi_k\left(\frac{\alpha}{n^{3/2}}\right)\right), \end{aligned}$$

where $\phi_k(\alpha)$ denotes the Laplace transform of $\mathcal{G}(p(n, k))$:

$$\phi_k(\alpha) = \frac{e^{-\alpha} p(n, k)}{1 - (1 - p(n, k))e^{-\alpha}}.$$

Recalling that $p(n, i) = \varepsilon(1 - \varepsilon)^i$ and $\varepsilon = c/n$, we find that, as $n \rightarrow \infty$,

$$\prod_{k=0}^{n-1} \left(1 + \phi_k\left(\frac{\alpha}{n^{3/2}}\right)\right) \simeq 2^n \exp\left(-\alpha \frac{\sqrt{n}}{2c^2} (e^c - 1)\right) \exp\left(\frac{3\alpha^2 (e^{2c} - 1)}{8c^3}\right).$$

Thus, substituting this into (3.9), we find that as $n \rightarrow \infty$,

$$E(\exp(-\alpha \Theta_n)) \simeq \exp\left(\frac{3\alpha^2 (e^{2c} - 1)}{8c^3}\right).$$

We recognize the Laplace transform of a centered Gaussian variable of variance $3(e^{2c} - 1)/8c^3$. This observation completes the proof.

4. Proof of Theorem 1.3

In the $(1 + 1)_L^*$ -EA framework, the algorithm may visit several distinct protein conformations (individuals) of the same fitness value before it finally jumps to a new individual with a higher fitness score. This is not allowed in the $(1 + 1)_L$ -EA, where the individual at time k , X_k , is not allowed to jump to another individual X_{k+1} with the same fitness score.

As in the previous sections, we denote by $(\tilde{X}_k)_k$ the chain defined by the protein conformations taken at times of fitness jumps $(\tau_k)_k$; i.e. $\tau_0 = 0$ and, for every $k \geq 0$,

$$\tau_{k+1} = \inf\{i \geq \tau_k : L(X_i) \neq L(X_{\tau_k})\}, \quad \sigma_{k+1} = \tau_{k+1} - \tau_k, \quad \tilde{X}_k = X_{\tau_k}.$$

Now, we briefly sketch the proof of Theorem 1.3.

4.1. Proof in the one-flip case

Here, we put $\varepsilon = 1/n$ and keep the definition for $(W_k)_k$ that was used in Section 3. In the $(1 + 1)_L$ -EA framework, once X_0 has been sampled, the path $(\tilde{X}_k)_k$ becomes entirely deterministic: going along it means flipping the 0s of X_0 one at a time, from left to right. This is not true in the case of the $(1 + 1)_L^*$ -EA, and we cannot directly adapt the proof of Theorem 1.1. However, the method used in the proof of Theorem 1.2 is valid. First, we consider the law of \tilde{T}_n conditioned on the values taken by L , along the path $(\tilde{X}_k)_k$, until the target individual is hit, and we give the analog of Lemma 3.3 (see Lemma 4.1). Then, we show that the other key argument in the proof of Theorem 1.2, which is contained in Lemma 3.4 and its direct application, Lemma 3.5, is still valid (see Lemma 4.2). That is, if X_0 is chosen uniformly and

at random in $\{0, 1\}^n$ then, at the time of each fitness jump τ_k , whatever the LeadingOnes score at time τ_{k-1} (say $\ell_{k-1} = i_0$), the result of the k th jump on block Y_{k-1} , i.e. block W_k , remains uniformly distributed in $\{0, 1\}^{n-i_0-1}$.

Lemma 4.1. *Conditioned on $\{\ell_0 = i_0, \dots, \ell_{J-1} = i_{J-1}, \ell_J = n\}$, \widehat{T}_n is the sum of J independent, identically distributed geometric random variables with parameter ε , i.e. \widehat{T}_n is negative binomial of parameter (J, ε) .*

Proof. This is a straightforward adaptation of the proof of Lemma 3.3. In the one-flip case, Lemma 3.2 still holds and Lemma 3.1 is replaced by the fact that, conditioned on $\{\ell_0 = i_0, \dots, \ell_{J-1} = i_{J-1}, \ell_J = n\}$, the sequence $\{\sigma_k, k = 1, \dots, J\}$ is independent, identically distributed, and has law $\mathcal{G}(\varepsilon)$.

Lemma 4.2. *Let $k \geq 1$. If X_0 is chosen uniformly in $\{0, 1\}^n$ then, given that $\{\ell_{k-1} = i\}$ occurs, W_k follows the uniform distribution on $\{0, 1\}^{n-i-1}$, i.e.*

$$\mathcal{L}(W_k \mid \ell_{k-1} = i) = \mathcal{U}(\{0, 1\}^{n-i-1}).$$

Proof. The proof is identical to that of Lemma 3.4, up to (3.4):

$$P_i(W_1 = w) = \frac{P_i(X_1 = (1^i, 1, w))}{P_i(L(X_1) > L(X_0))}. \tag{4.1}$$

In the one-flip scenario, a single unique bit is flipped during the mutation step. Therefore, in order to sample $(1^i, 1, w)$ at time 1, we must sample $(1^i, 0, w)$ at time 0. Thus,

$$P_i(X_1 = (1^i, 1, w)) = P(X_1 = (1^i, 1, w) \mid X_0 = (1^i, 0, w)) P_i(Y_0 = w) = 1/n 2^{n-i-1}. \tag{4.2}$$

On the other hand, as the fitness score is increased if the left-most 0 of the chain is flipped,

$$P_i(L(X_1) > L(X_0)) = 1/n. \tag{4.3}$$

Hence, by applying (4.2) and (4.3) to (4.1), we recover the result for $k = 1$. Finally, the strong Markov property gives the result for $k > 1$, completing the proof.

Now we can prove Theorem 1.3 in the one-flip scenario.

Proof of Theorem 1.3. The proof follows that of Theorem 1.2, in which Lemma 3.2 remains valid but Lemmas 4.1 and 4.2 replace Lemmas 3.3 and Lemma 3.5. We have

$$\begin{aligned} P(\widehat{T}_n = t) &= \frac{1}{2^n} \sum_J \sum_{o \leq i_0 \leq i_1 \leq \dots \leq i_J = n} \mathcal{NB}(J, \varepsilon)(t) \\ &= \sum_J \frac{1}{2^n} \binom{n}{J} \mathcal{NB}(J, \varepsilon)(t). \end{aligned}$$

On the other hand, we recall from Lemma 2.1 that, conditioned on $|X_0| = n - J$, T_n is negative binomial of parameter (J, ε) . Thus, as X_0 is chosen uniformly and at random in $\{0, 1\}^n$, it yields

$$\begin{aligned} P(T_n = T) &= \sum_J P(T_n = t \mid |X_0| = n - J) P(|X_0| = n - J) \\ &= \sum_J \mathcal{NB}(J, \varepsilon)(t) \frac{1}{2^n} \binom{n}{J}, \end{aligned}$$

and we find that

$$P(T_n = t) = P(\widehat{T}_n = t),$$

for all $t \geq 0$.

4.2. Proof in the Bernoulli-flip case

The dynamics of the $(1 + 1)_L^*$ -EA and the $(1 + 1)_L$ -EA differ slightly. Nevertheless, we notice that in the $(1 + 1)_L^*$ -EA framework, the proofs of Lemmas 3.1, 3.2, 3.4, and 3.5 still hold. Now, since the proof of Proposition 3.1 is entirely based on these lemmas, we conclude that the probability distribution of the hitting time is the same in both the $(1 + 1)_L^*$ -EA and the $(1 + 1)_L$ -EA scenarios.

5. Conclusion

After examining the two versions (one-flip and Bernoulli-flip) of the EAs on which we have focused, we have found that, as $(e^c - 1)/c^2 > 1$ for all $c \in \mathbb{R}^+$, the expected value of the hitting time is higher in the Bernoulli-flip case than it is in the one-flip case. Thus, we can conclude that the one-flip performs better than any Bernoulli-flip, in terms of the expected hitting time. The same conclusion has already been derived by Garnier *et al.* [13] for the OneMax problem.

The superior performance of the one-flip suggests that, in the LeadingOnes framework, despite the ability of the Bernoulli-flip to jump from any region of the search space to any other in a single iteration of the search process, the Bernoulli-flip results in a slower convergence to a given individual.

In order to explain this phenomenon, as the Markov chain that models our $(1 + 1)$ search process accepts a mutation only in the case of an increase in the number of leading 1s, we should point out the following facts. In the Bernoulli-flip framework, the closer the algorithm gets to the target individual, the longer the algorithm waits until it jumps; on the other hand, in the one-flip case, the number of leading 1s currently present in the bit-string does not affect the distribution of the time taken for the search to jump. Also, this probability distribution remains stable as the search converges on the optimal individual.

Acknowledgements

I wish to thank the anonymous referee for useful remarks. I am in debt to Jean Brard for introducing me to Garnier *et al.* [13] and for instructive discussions on evolution strategies. I also gratefully acknowledge Christian Mazza and Didier Piau for the improvements they suggested and their valuable comments.

References

- [1] ANFINSEN, C. B. (1973). Principles that govern the folding of a protein chain (Nobel lecture). *Science* **191**, 223–230.
- [2] BÄCK, T. (1993). Optimal mutation rates in genetic search. In *Proc. 5th Internat. Conf. Genet. Algorithms*, ed. S. Forrest, Morgan Kaufmann, San Francisco, pp. 2–8.
- [3] BAKK, A. *et al.* (2000). Pathways in two-state protein folding. *Biophys. J.* **79**, 2722–2727.
- [4] BÉRARD, J. AND BIENVENÛE, A. (2000). Convergence of a genetic algorithm with finite population. In *Mathematics and Computer Science* (Versailles, 2000), Birkhäuser, Basel, pp. 155–163.
- [5] BÉRARD, J. AND BIENVENÛE, A. (2003). Sharp asymptotic results for simplified mutation-selection algorithms. *Ann. Appl. Prob.* **13**, 1534–1568.
- [6] BINDER, K. (1987). *Applications of the Monte Carlo Method in Statistical Physics*. Springer, Berlin.
- [7] CERF, R. (1996). The dynamics of mutation-selection algorithms with large population sizes. *Ann. Inst. H. Poincaré Prob. Statist.* **32**, 455–508.

- [8] CERF, R. (1998). Asymptotic convergence of genetic algorithms. *Adv. Appl. Prob.* **30**, 521–550.
- [9] DEL MORAL, P. AND GUIONNET, A. (2001). On the stability of interacting processes with applications to filtering and genetic algorithms. *Ann. Inst. H. Poincaré Prob. Statist.* **37**, 155–194.
- [10] DILL, K. A., FEIBIG, K. M. AND CHAN, H. S. (1993). Cooperativity in protein folding kinetics. *Proc. Nat. Acad. Sci. USA* **90**, 1942–1946.
- [11] DROSTE, S., JANSEN, T. AND WEGENER, I. (1998). A rigorous complexity analysis of the (1 + 1) evolutionary algorithm for linear functions with boolean inputs. In *Proc. IEEE Internat. Conf. Evolutionary Comput.* (ICEC'98). IEEE Press, Piscataway, NJ, pp. 499–504.
- [12] DROSTE, S., JANSEN, T. AND WEGENER, I. (2002). On the analysis of the (1 + 1) evolutionary algorithm. *Theoret. Comput. Sci.* **276**, 51–81.
- [13] GARNIER, J., KALLEL, L. AND SCHOENAUER, M. (1999). Rigorous hitting times for binary mutations. *Evolutionary Comput.* **7**, 173–203.
- [14] JANSEN, T. AND WEGENER, I. (2001). How to cope with plateaus of constant fitness and when to reject strings of the same fitness. *IEEE Trans. Evolutionary Comput.* **5**, 589–599.
- [15] LEVINTHAL, C. (1968). Are there pathways for protein folding? *J. Chim. Phys. Phys.-Chim. Biol.* **65**, 44–45.
- [16] MAZZA, C. AND PIAU, D. (2001). On the effect of selection in genetic algorithms. *Random Structures Algorithms* **18**, 185–200.
- [17] MÜHLENBEIN, H. (1992). How genetic algorithms really work: I. Mutation and hill-climbing. In *Parallel Problem Solving from Nature PPSN II*, eds R. Männer and R. Manderick, North-Holland, Amsterdam, pp. 15–25.
- [18] RABINOVICH, Y. AND WIGDERSON, A. (1999). Techniques for bounding the convergence rate of genetic algorithms. *Random Structures Algorithms* **14**, 111–138.
- [19] RUDOLPH, G. (1997). *Convergence Properties of Evolutionary Algorithms*. Kovac, Hamburg.
- [20] SCHELLMAN, J. A. (1958). The factors affecting the stability of hydrogen-bounded polypeptide structures in solution. *J. Phys. Chem.* **62**, 1485–1494.