

# Runtime Analysis of Non-Elitist Evolutionary Algorithms with Fitness-Proportionate Selection on Royal Road Functions

Anton V. Eremeev

Sobolev Institute of Mathematics SB RAS,  
Omsk, Russia,

The Institute of Scientific Information  
for Social Sciences RAS, Moscow, Russia,  
E-mail: eremeev@ofim.oscsbras.ru

**Abstract**—It has been shown before that some results from the theory of evolutionary algorithms (EAs) may be used for the analysis of population dynamics in biology. In the present paper, we study the EAs without elite individuals, where the fitness-proportionate selection is used, on the class of Royal Road fitness functions. The behavior of EAs with these fitness functions may be considered as a model of population dynamics in the case of fitness landscape with several neutrality regions. We obtain lower and upper bounds on the expected time of evolving the fittest possible genotype for the Royal Road fitness function. It is expected the obtained results to be useful in development of modeling tools for biological populations.

**Index Terms**—Optimization time, Royal Road function, Fitness proportionate selection, Scaling

## I. INTRODUCTION

It has been shown before that some results from the theory of evolutionary algorithms (EAs) may be used for the analysis of population dynamics in biology (see e.g. [1]). In this paper, we study non-elitist EAs with fitness-proportionate selection on the class of Royal Road fitness functions.

In the theory of EAs, the term *runtime*, or *optimization time* stands for the number of individuals produced and evaluated, until a genotype with the maximum fitness is found for the first time. Here evaluation of an individual usually implies computing the fitness of its genotype, which is equivalent to finding the objective function value of a tentative solution to the optimization problem being solved by the evolutionary algorithm. Results of the EAs theory describe the role of mutation and crossover operators, population size, or self-adaptation methods etc., see e.g. [2]–[7]. From studies of such kind, some improved evolutionary algorithms and their operators may be developed [8], [9] and some biologically meaningful results may be obtained [1], [10], [11].

In particular, in [1], [10] the theoretical bounds known for the EAs with a variant of the truncation selection called  $(\mu, \lambda)$ -selection, were applied to model the genetic engineering technique of Systematic Evolution of Ligands by EXponential enrichment (SELEX). The upper bounds on the expected EA runtime were used to upper-bound the expected number of rounds of SELEX until a series of binding sites

for protein factors is found. It was shown [10] that for some cases with large population size, theoretical bounds give favorable prediction, while computational experiments require prohibitive CPU resource (e.g. with populations of  $10^{14}$  or  $10^{15}$  individuals). In [11], it is demonstrated that an EA with tournament selection, applied to a string matching problem, may be used as a meaningful simulation model for some SELEX procedures with populations up to  $10^8$  individuals.

### A. Preliminaries

To describe the EA outline, let us consider a problem of maximizing of a function  $f(x)$  over the set of all  $x \in \mathcal{A}^n$ , where  $\mathcal{A}$  is a finite alphabet which is used for solutions encoding. In the EA literature, the objective function  $f$  is called a *fitness function*, and  $\mathcal{A}^n$  is called the space of genotypes. In what follows, we assume that  $\mathcal{A} = \{0, 1\}$  for the sake of simplicity. A population of  $\lambda$  individuals on the EA iteration  $t$  is denoted by  $P_t$ .

$$P_t = (P_t(1), \dots, P_t(\lambda)) \in \mathcal{A}^{n\lambda},$$

where  $P_t(k)$  is the  $k$ -th individual in  $P_t$ ,  $k = 1, \dots, \lambda$ .

The scheme of the EA considered in this paper is as follows.

#### Algorithm 1

1. Construct a random initial population  $P_0$ .
2. Put  $t := 0$ .
2. Until a stopping condition is met, repeat steps 2.1 and 2.2:
  - 2.1. Put  $t := t + 1$ .
  - 2.2. For  $k := 1$  to  $\lambda$  repeat steps 2.2.1 and 2.2.2:
    - 2.2.1. Apply proportionate selection to choose a genotype  $y$  in  $P_t$ .
    - 2.2.2. Add a genotype  $\text{mutate}(y)$  to population  $P_{t+1}$ .
3. Output the the fittest genotype in  $P_0, \dots, P_{t+1}$ .

We assume that the mutation operator  $\text{mutate}$  changes each bit of  $x$  with a given mutation probability  $\chi/n$ , where the parameter  $\chi$  is usually a constant. Roughly speaking, the runtime of such an EA is  $\lambda$  times the number of iterations till the optimal genotype is found for the first time.

In the evolutionary computation, one of the well-known examples of the fitness function, generalizing the linear functions, is the Royal Road function [12]. In what follows, we use the most studied special case of these functions as in [13]. Here a string  $x$  is partitioned into  $n/r$  blocks of length  $r$  (in what follows, we assume the parameter  $r$  to be a constant):

$$\text{RR}_r(x) := \sum_{i=0}^{n/r-1} a_i \prod_{j=1}^r x_{ir+j}.$$

A number of well-known results in the theoretical biology are based on the assumption of weak epistasis, which implies that genes effect on the fitness of a genotype is approximately additive [11], [14]. The Royal Road functions may be considered as example of such functions, where each block corresponds to a separate gene or domain, and there are a number of such blocks in the genotype. A modification of the RR function for the case of four-letter alphabet  $\mathcal{A}$  may be found in [1]. In the most basic model of additive effects, one can consider the linear fitness functions, which may be considered as a special of the Royal Road functions with  $r = 1$ .

In this paper, we can assume w.l.o.g. that the weights  $a_j$  are positive and ordered so that  $a_1 \geq a_2 \geq \dots \geq a_n > 0$ . The behavior of EAs on the fitness functions  $\text{RR}_r(x)$  may be considered as a simplified model of population dynamics in the presence of neutrality regions in a biological fitness landscape [15].

Let us introduce some other notations and definitions which will be used in what follows.

If  $\mathcal{X}$  is partitioned into  $m$  subsets, called *levels* ( $A_1, \dots, A_m$ ), let us denote  $A_{\geq j} := \cup_{i=j}^m A_i$ . A partition is called an *f-based* partition if for any  $x \in A_j, y \in A_{j+1}$  we have  $f(y) > f(x)$ ,  $j = 1, \dots, m-1$ . With a minor abuse of the standard notation, to denote the number of individuals of  $P$  in  $A \subseteq \mathcal{X}$  we use  $|P \cap A| := |\{k : P(k) \in A\}|$ .

Formally, the selection operator is represented by a probability distribution over  $1, \dots, \lambda$ , and we use  $p_{\text{sel}}(i | P)$  to denote the probability of selecting the  $i$ -th individual  $P(i)$ . In the proportionate selection, the chance of selecting an individual is equal to its fitness divided by the sum of fitness values in the current population:

$$p_{\text{sel}}(i | P) = \frac{f(P(i))}{\sum_{j=1}^{\lambda} f(P(j))}.$$

In this paper, w.l.o.g we assume that individuals of  $P$  are sorted so that  $f(P(i)) \geq f(P(i+1))$ ,  $i = 1, \dots, \lambda - 1$ . The *cumulative selection probability*  $\beta$  for any  $\gamma \in (0, 1]$  is defined [16] as

$$\beta(\gamma, P) := \sum_{i=1}^{\lambda} p_{\text{sel}}(i | P) \cdot [f(P(i)) \geq f_{\lceil \gamma \lambda \rceil}],$$

where  $P \in \mathcal{X}^{\lambda}$ , and  $[\cdot]$  stands for the Iverson bracket. One can interpret  $\beta(\gamma, P)$  as the probability to select an individual with fitness at least as high as in the  $\lceil \gamma \lambda \rceil$ -ranked individual of  $P$ .

The mutation operator `mutate` is represented by a transition matrix  $p_{\text{mut}}$  with  $|\mathcal{X}|^2$  elements, where  $p_{\text{mut}}(y | x)$  is the probability to mutate an individual  $x$  into  $y$ .

The following Level-Based Theorem is a re-formulation of Corollary 7 [17], adjusted to the  $f$ -based partition and Algorithm 1, and improving Theorem 8 of [18].

**Theorem 1.** *Assuming an  $f$ -based partition  $(A_1, \dots, A_m)$  of  $\mathcal{X}$ , and  $T := \min\{t\lambda \mid |P_t \cap A_m| > 0\}$ , if there exist parameters  $s_1, \dots, s_{m-1}, p_0, \delta \in (0, 1]$ ,  $\gamma_0 \in (0, 1)$  such that*

**(M1)** *for all  $P \in \mathcal{X}^{\lambda}$  for all  $j = 1, \dots, m-1$  holds*

$$p_{\text{mut}}(y \in A_{\geq j+1} \mid x \in A_j) \geq s_j,$$

**(M2)** *for all  $P \in \mathcal{X}^{\lambda}$ , for all  $j \in [m-1]$  holds*

$$p_{\text{mut}}(y \in A_{\geq j} \mid x \in A_j) \geq p_0,$$

**(M3)** *for all  $P \in (\mathcal{X} \setminus A_m)^{\lambda}$ , for all  $\gamma \in (0, \gamma_0]$  holds*

$$\beta(\gamma, P) \geq (1 + \delta)\gamma/p_0,$$

**(M4)** *population size  $\lambda$  satisfies*

$$\lambda \geq \frac{4}{\gamma_0 \delta^2} \ln \left( \frac{128m}{\gamma_0 s_* \delta^2} \right), \text{ where } s_* := \min_{j=1, \dots, m-1} \{s_j\},$$

*then  $\mathbf{E}[T] < \left(\frac{8}{\delta^2}\right) \sum_{j=1}^{m-1} \left(\lambda \ln \left(\frac{6\delta\lambda}{4+\gamma_0 s_j \delta\lambda}\right) + \frac{1}{\gamma_0 s_j}\right)$ .*

## B. Some Previous Results

The fitness-proportionate selection, which is also called a *roulette-wheel* selection, was most frequently used in the first genetic algorithms [19]. Unlike the rank-based selections [16] (tournament selection,  $(\mu, \lambda)$ -selection, ranking selection etc.), the probability distribution of this operator explicitly depends on the absolute values of the individuals fitness. Thus, a composition of fitness with some non-linear function may significantly change this distribution.

The field of evolutionary computation emerged due to transfer of ideas from the evolutionary biology into the computer science. Currently, a reverse transfer of some methods from the evolutionary computation can be observed (see e.g. [20]). Studies of the proportionate selection may also contribute to this reverse transfer, since several well-known models of population dynamics in the population genetics also express the fitness-proportionate effect of selection on the genotypes frequencies [21]. (Note that in the population genetics, the fitness is typically defined as the expected number of offspring of an individual or the underlying genotype, which corresponds to the term ‘‘reproductive rate’’ in the case of EAs.)

The assumption that the genes effects on the fitness are approximately additive is used in many models in the theoretical biology, however sometimes it is more appropriate to assume that genes have multiplicative effects [14]. Counterparts of biological models with multiplicative effects can be found in the EAs, based on the proportionate selection *with fitness scaling* [16], [22]. On the most basic linear function  $\text{ONEMAX}(x) := \sum_{i=1}^n x_i$ , Lehre [16] showed that the EA with a sufficiently large population requires an expected exponential time to find a bit-string with more than 99.7% of 1-bits. On the other hand, Neumann et al. [22] proved that a scaling of the original fitness function, which is equivalent to changing it into  $(\lambda - 1)^{\text{ONEMAX}(x)}$  with  $\lambda > c \log n$ , for some constant  $c > 0$  guarantees expected optimization time  $O(\lambda n)$ .

### C. Our Contribution

The results presented here are based on the methods suggested in the preprint [23]. The contribution is twofold. On one hand, in the next section it is shown that even the linear functions require an exponentially growing in  $n$  amount of time to find the optimum, if the non-elitist EA uses the fitness-proportionate selection, the bitwise mutation with a mutation probability  $\chi/n$ , and a population of size at least  $n^k$ , where  $\chi > \ln 2$  and  $k > 2$  are constants. On the other hand, by application of the Level-Based Theorem [17] in Section III we show that in the case of constant block length  $r$  and integer polynomially-bounded weights  $a_i$ , the EA has a polynomially bounded runtime, given the appropriate parameter  $\chi$ , inversely proportional to  $n$  and a sufficiently large but polynomially bounded  $\lambda$ . We also show in Section IV that polynomial expected runtime can be achieved using a constant  $\chi$  if the fitness is exponentially scaled.

### II. INEFFICIENCY OF STANDARD MUTATION PROBABILITY

In this section, we consider Algorithm 1 with a constant value of the parameter  $\chi > \ln 2$ . To show that this algorithm is inefficient on linear fitness functions, we use the method proposed in [16] for the special case of the ONEMAX function.

**Theorem 2.** *If  $f(x) \equiv \sum_{i=1}^n a_i x_i$  on  $\mathcal{X}$ , then Algorithm 1 with population size  $\lambda \geq n^{2+\delta}$ , and a mutation probability  $\chi/n$  for any constants  $\delta > 0$ ,  $\chi > \ln(2)$ , obtains the optimum of  $f$  with probability at most  $\lambda e^{-c'n^\delta}$  within  $e^{cn}$  generations, where  $c, c'$  are some positive constants.*

The only difference of this formulation from that of Theorem 4 [23] is that here we do not require  $\lambda$  to be polynomially bounded. Theorem 2 is proved by means of the Negative Drift Theorem for Populations from [24]. The proof is analogous to that of Theorem 4 [23].

### III. EFFICIENCY OF LOW MUTATION PROBABILITIES

Authors of early computational experiments with the genetic algorithms (e.g. see [19]) recommended that the mutation probability should be chosen inversely proportional to the population size. Later, theoretical runtime analysis of simple EAs on the benchmark functions like ONEMAX indicated that the mutation probability should be inversely proportional to the problem dimension (see e.g. [25]). However, as the previous section shows, a constant value of  $\chi$  should not be recommended in the case of fitness-proportionate selection on linear functions. A similar conclusion was obtained from the analysis of genetic algorithms [26]–[28] in the special case of ONEMAX. Later, in [18] it was discovered that reducing the mutation probability to  $1/(6n^2)$  one gets a polynomial expected runtime on ONEMAX. Now we generalize the result from [18] to the Royal Road functions. In this section, we assume that weights  $a_j$  to be integer numbers.

**Theorem 3.** *If  $f(x) \equiv \text{RR}(x)$  on  $\mathcal{X}$ , then Algorithm 1, using mutation probability  $\chi/n$  with  $\chi = (1-c)/(na_{\max})$ ,  $a_{\max} := \max_{i=1}^n a_i$ , for any positive constant  $c < 1$ , and the*

*size of population  $\lambda \geq Cn^2 a_{\max}^2 r \ln(na_{\max})$  for a sufficiently large constant  $C > 0$ , has the expected runtime no more than*

$$\frac{2^7 n^3 a_{\max}^2}{c^2} \left( \lambda \ln \lambda + \frac{4en^{2r} a_{\max}^r}{c(1-c)^r} \right). \quad (1)$$

*Proof.* Let us denote  $N := n/r$ . We assume w.l.o.g. that  $a_1 \geq a_2 \geq \dots \geq a_N > 0$  in order to apply Theorem 1, partitioning the search space into the following levels:

$$A_j := \left\{ x \mid \sum_{i=1}^j a_i \leq \text{RR}(x) < \sum_{i=1}^{j+1} a_i \right\}, \quad j = 0, \dots, N-1, \\ A_N := \{x \mid \text{RR}(x) = f^*\},$$

where  $f^* = \sum_{\ell=1}^N a_\ell$ . In total we have  $m = N + 1$  levels.

Let us say for any  $x \in \mathcal{X}$  that a block  $\ell$  is solved in  $x$  if  $x_{i(\ell)} = \dots = x_{i(\ell)+r} = 1$ ,  $i(\ell) := (\ell-1)r + 1$ .

By the definition of our partition, if  $x \in A_j$  then for any  $j < N$ , there is at least one block  $\ell$  among the blocks  $1, \dots, j+1$  that is not solved. So it is sufficient to solve block  $\ell$  and keep the rest of the bits unchanged in order to get an individual at a level above  $j$ . This happens with a probability not less than  $(1 - \chi/n)^{n-r} (\chi/n)^r$ , which may be lower-bounded by

$$\left(1 - \frac{1}{n}\right)^{n-1} \left(\frac{1-c}{a_1 n^2}\right)^r \geq \frac{(1-c)^r}{ea_1^r n^{2r}},$$

so if we put  $s_j = s_* := ((1-c)^r)/(ea_1^r n^{2r})$ ,  $j = 1, \dots, m$ , condition (M1) will be satisfied.

We put  $p_0 := (1 - \chi/n)^n$ , which is the probability to keep a solution unchanged in mutation. So (M2) is satisfied.

To satisfy (M3), we first put  $\gamma_0 := c/4$ . The fitness of the  $\lceil \gamma \lambda \rceil$ -ranked individual for any  $\gamma \leq \gamma_0$  we denote for brevity  $f_\gamma$ . Then at least  $k \geq \lceil \gamma \lambda \rceil \geq \gamma \lambda$  individuals have a fitness greater or equal to  $f_\gamma$ . The total fitness of these individuals is  $q \geq k f_\gamma \geq \gamma \lambda f_\gamma$ . In view of integrality of all  $a_i$ , we can make a worst-case assumption that all individuals less fit than  $f_\gamma$  are of fitness  $f_\gamma - 1$ . Thus we have

$$\begin{aligned} \beta(\gamma, P) &\geq \frac{q}{q + (\lambda - k)(f_\gamma - 1)} \geq \frac{q}{q + (\lambda - \gamma \lambda)(f_\gamma - 1)} \\ &\geq \frac{\gamma \lambda f_\gamma}{\gamma \lambda f_\gamma + (\lambda - \gamma \lambda)(f_\gamma - 1)} = \frac{\gamma}{1 - (1 - \gamma)/f_\gamma} \\ &\geq \frac{\gamma}{1 - (1 - c/4)/f^*} \geq \gamma e^{(1-c/4)/f^*}. \end{aligned}$$

Here the last inequality follows from the fact that  $e^{-z} \geq 1 - z$ .

Note that  $p_0 = (1 - \chi/n)^n \geq \exp(-\chi(1 - \varepsilon)^{-1})$  for any positive constant  $\varepsilon < 1$  and  $n$  large enough. This can be shown as follows. We have  $e^{-y} = 1 - y + \alpha(y)y$ , where  $\alpha(y)$  tends to 0 when  $y \rightarrow 0$ . Thus for a positive  $\varepsilon$  and small enough  $y > 0$  we have  $e^{-y} \leq 1 - (1 - \varepsilon)y$ . Given a positive  $\varepsilon < 1$ , let us put  $y := \chi/(n(1 - \varepsilon))$ . Then  $(1 - \chi/n)^n \geq e^{-y^n} = \exp(-\chi/(1 - \varepsilon))$  for all  $n$  large enough.

So we conclude that

$$p_0 \beta(\gamma, P) \geq \left( \frac{1 - f^* \chi / (1 - \varepsilon) - c/4}{f^*} + 1 \right) \gamma.$$

Since  $f^*\chi \leq \chi Na_1 \leq \chi na_1 = 1 - c$ , so by setting  $\varepsilon := 1 - \frac{1-c}{1-c/2}$  we get  $f^*\chi(1-\varepsilon)^{-1} \leq 1 - c/2$  and

$$p_0\beta(\gamma, P) \geq \gamma \left(1 + \frac{c}{4na_1}\right),$$

so condition (M3) holds for  $\delta := c/(4na_1)$ .

To satisfy condition (M4), we need a population size to be at least

$$\begin{aligned} & \frac{4}{\gamma_0\delta^2} \ln \left( \frac{128m}{\gamma_0 s_* \delta^2} \right) \\ &= \frac{16}{c(c/(4na_1))^2} \ln \left( \frac{512(N+1)}{c((1-c)^r/(en^{2r}a_1^r))(c/(4na_1))^2} \right) \\ &< \frac{2^8 n^2 a_1^2}{c^3} \left( \ln(N+1) + (2r+2) \ln n \right. \\ &\quad \left. + (r+2) \ln a_1 + 11 + \ln \frac{1}{c(1-c)^r} \right) \\ &\leq Cn^2 a_1^2 r \ln(na_1) \text{ for a sufficiently large constant } C. \end{aligned}$$

The condition then holds for any  $\lambda \geq Cn^2 a_1^2 r \ln(na_1)$ .

Since all the conditions are satisfied, application of Theorem 1 implies

$$\begin{aligned} \mathbf{E}[T] &\leq \frac{8}{\delta^2} \sum_{j=1}^N \left( \lambda \ln(3\delta\lambda/2) + \frac{1}{\gamma_0 s_j} \right) \\ &\leq \frac{2^7 n^3 a_1^2}{c^2} \left( \lambda \ln \lambda + \frac{4en^{2r}a_1^r}{c(1-c)^r} \right). \end{aligned}$$

□

Theorem 3 in particular implies that if  $a_i = 1$  for all  $i = 1, \dots, n$ , Algorithm 1 has a polynomially bounded runtime, if the parameters  $\chi$  and  $\lambda$  are chosen appropriately. For sufficiently large  $\lambda$ , the runtime bound (1) can be asymptotically improved using Theorem 3.2 from [29] instead of Theorem 1. The formulation of Theorem 3.2 in [29] however does not provide explicitly the constants hidden in  $\mathcal{O}(\cdot)$ -notation.

#### IV. EFFICIENCY OF EXPONENTIAL FITNESS SCALING

In this section, we show that the EA can perform efficiently also if one applies an appropriate nonlinear scaling to the fitness function. Again we assume all weights  $a_j$  to be integer numbers.

Neumann et al [22] proposed the following modification of the fitness function, which may be called *exponential fitness scaling*:

$$f(x, s) := s^{f(x)}, \quad (2)$$

where  $s$  is a tunable parameter. In [22], it was proved that the EA can efficiently optimize ONEMAX and some other similar functions, assuming  $s := \lambda - 1$ , which grows at least logarithmically in  $n$ . In the following theorem, it is sufficient that  $s > e^\chi$  is just a constant.

**Theorem 4.** *If  $f(x) \equiv s^{\text{RR}(x)}$  on  $\mathcal{X}$ , where  $s > e^\chi$ , then Algorithm 1, using mutation probability  $\chi/n$  with a constant*

$\chi > 0$ , and the size of population  $\lambda \geq \frac{4s}{\varepsilon^3} \ln \left( \frac{128(n/r+1)sen^r}{\varepsilon^3 \chi^r} \right)$ , where  $\varepsilon := \sqrt[3]{\frac{s}{e^\chi}} - 1$ , has the expected runtime at most

$$\frac{8n}{\varepsilon^2 r} \left( \lambda \ln(3\varepsilon\lambda/2) + \frac{n^r es}{\varepsilon \chi^r} \right) = \mathcal{O}(n\lambda \ln \lambda + n^{r+1}).$$

The proof is analogous to that of Theorem 3. The main difference consists in the application of the Level-Based Theorem, where the parameter  $\delta$  now can be set to a constant. In other words, the selection mechanism behaves similarly to a rank-based selection [17].

*Proof.* We apply Theorem 1, using the same set of levels as in Theorem 3. The total number of levels is  $m = N + 1$ , where  $N = n/r$ . We have a lower bound for the improvement probability  $s_* = s_j := \left(1 - \frac{\chi}{n}\right)^{n-r} \left(\frac{\chi}{n}\right)^r \geq \frac{1}{e} \left(\frac{\chi}{n}\right)^r$ , which is suitable for condition (M1).

To meet condition (M2), let us set  $p_0 := (1 - \chi/n)^n$ , i.e. the probability of keeping the solution unchanged in mutation.

To verify (M3), let us put  $\gamma_0 := \varepsilon/s$ . Now  $\chi$  as well as  $s$  are constants and  $s > e^\chi$ , therefore  $\varepsilon$  and  $\gamma_0$  are constants, and  $\gamma_0 \in (0, 1)$ .

Let us denote the level of a  $\lceil \gamma\lambda \rceil$ -ranked individual by  $f_j$ . Assume that  $k \geq \lceil \gamma\lambda \rceil$  is the number of individuals at least as fit as  $\lceil \gamma\lambda \rceil$ -ranked individual. The total fitness of these  $k$  individuals, denoted by  $q$ , satisfies the inequality  $q \geq kf_j \geq \lceil \gamma\lambda \rceil f_j$ . For the rest of the individuals we have  $f(x, s) \leq f_j/s$  since  $f(x)$  is integer, given integer  $a_1, \dots, a_N$ . The probability to select one of the  $k$  individuals is

$$\beta(\gamma, P) \geq \frac{\gamma}{(1 - \frac{\chi}{n})/s + \gamma} \geq \frac{\gamma s}{1 + \varepsilon}$$

for any  $\gamma \leq \gamma_0 = \varepsilon/s$ .

The lower bound for  $p_0 := (1 - \chi/n)^n$ , as it was shown in the proof of Theorem 3, gives  $p_0 \geq e^{-\chi}/(1 + \varepsilon)$ , where  $\varepsilon > 0$  is a constant and  $n$  is large enough. So it holds that

$$p_0\beta(\gamma, P) \geq \frac{\gamma s}{e^\chi(1 + \varepsilon)^2} = \gamma(1 + \varepsilon).$$

The last equality holds because  $s = e^\chi(1 + \varepsilon)^3$  due to the choice of  $\varepsilon$ . Therefore condition (M3) holds with a constant  $\delta := \varepsilon$ .

According to (M4), the population size should be not less than  $\frac{4}{\gamma_0\delta^2} \ln \left( \frac{128m}{\gamma_0 s_* \delta^2} \right)$ . The condition then holds for any  $\lambda \geq \frac{4s}{\varepsilon^3} \ln \left( \frac{128(n/r+1)sen^r}{\varepsilon^3 \chi^r} \right)$ . Since all the conditions are satisfied, application of Theorem 1 implies

$$\begin{aligned} \mathbf{E}[T] &\leq \frac{8}{\delta^2} \sum_{j=1}^{n/r} \left( \lambda \ln(3\delta\lambda/2) + \frac{n^r e}{\gamma_0 \chi^r} \right) \\ &= \frac{8n}{\varepsilon^2 r} \sum_{j=1}^{n/r} \left( \lambda \ln(3\varepsilon\lambda/2) + \frac{n^r es}{\varepsilon \chi^r} \right). \end{aligned}$$

□

## V. CONCLUSIONS AND FURTHER RESEARCH

The paper presents runtime analysis of fitness-proportionate selection on the class of Royal Road fitness functions and the linear functions as its subclass. It is supposed that the obtained lower and upper bounds may be further elaborated to deal with four-letter alphabet, thus yielding results meaningful for biology, where the DNA encoding is based on four letters.

On one hand, we claim that with an overwhelming probability the non-elitist EAs using fitness-proportionate selection and the bitwise mutation have at least exponential runtime on the whole class of linear fitness functions, if a mutation probability  $\chi/n$  with a constant value of parameter  $\chi > \ln 2$  is used. On the other hand, we demonstrate that the expected runtime can be upper-bounded by a polynomial on any Royal Road function if this function is polynomially bounded and the parameter  $\chi$  is inversely proportional to the string length. Alternatively, polynomially bounded expected runtime may be achieved in the case of exponential scaling of the Royal Road fitness function.

Further research might improve the upper bounds on the runtime of the considered EA, using new versions of the Level Based Theorem, tighter than the one obtained in [17]. This may be possible, e.g. if the constants in the asymptotic expressions are found explicitly in the approach proposed in [29]. Another open question is to tighten the lower bounds on improvement probabilities  $s_j$ . Currently these lower bounds are chosen very pessimistically, assuming that if a Royal Road block is not chosen optimally, then all of its positions have wrong values. Obviously, in practice, most often this is not true. However in order to establish tighter lower bounds on improvement probabilities it is required to perform a deeper problem-specific analysis of the evolution of genotype substrings in the blocks of the Royal Road functions.

## ACKNOWLEDGEMENT

The author is grateful to Duc-Cuong Dang, Per Kristian Lehre and Alexander Spirov for helpful discussions. The research was supported by the Russian Science Foundation, grant number 17-18-01536.

## REFERENCES

- [1] A. Eremeev and A. Spirov, "Estimates from evolutionary algorithms theory applied to gene design," in *Proc. of 11th International Multiconference Bioinformatics of Genome Regulation and Structure \ Systems Biology*. IEEE, 2018, pp. 33–38.
- [2] D. Dang, T. Friedrich, T. Kotzing, M. S. Krejca, P. K. Lehre, P. S. Oliveto, D. Sudholt, and A. M. Sutton, "Escaping local optima using crossover with emergent diversity," *IEEE Trans. on Evolutionary Computation*, vol. 22, no. 3, pp. 484–497, June 2018.
- [3] D.-C. Dang, T. Jansen, and P. K. Lehre, "Populations can be essential in tracking dynamic optima," *Algorithmica*, vol. 78, no. 2, pp. 660–680, 2017.
- [4] D.-C. Dang and P. K. Lehre, "Self-adaptation of mutation rates in non-elitist populations," in *Proc of PPSN'16*, 2016, pp. 803–813.
- [5] T. Jansen and I. Wegener, "On the utility of populations in evolutionary algorithms," in *Proc. of GECCO'01*, 2001, pp. 1034–1041.
- [6] P. K. Lehre and E. Özcan, "A runtime analysis of simple hyper-heuristics: To mix or not to mix operators," in *Proc. of FOGA'13*, 2013, pp. 97–104.
- [7] C. Witt, "Population size versus runtime of a simple evolutionary algorithm," *Theor. Comput. Sci.*, vol. 403, no. 1, pp. 104–120, 2008.

- [8] B. Doerr, C. Doerr, and F. Ebel, "From black-box complexity to designing new genetic algorithms," *Theor. Comput. Sci.*, vol. 567, pp. 87–104, 2015.
- [9] B. Doerr, H. P. Le, R. Makhmara, and T. D. Nguyen, "Fast genetic algorithms," in *Proc. of GECCO'17*, 2017, pp. 777–784.
- [10] A. Eremeev and A. Spirov, "Evaluation of runtime bounds for selex procedure with high selection pressure," in *Proceedings of the GECCO Companion*, ser. 19. New York, NY, USA: ACM, 2019, pp. 113–114.
- [11] I. S. Oh, Y. Lee, and R. McKay, "Simulating chemical evolution," in *2011 IEEE Congress of Evolutionary Computation (CEC)*, 2011, pp. 2717–2724.
- [12] M. Mitchell, S. Forrest, and J. H. Holland, "The royal road for genetic algorithms: Fitness landscapes and ga performance," in *Proceedings of the First European Conference on Artificial Life*. MIT Press, 1991, pp. 245–254.
- [13] T. Storch and I. Wegener, "Real royal road functions for constant population size," *Theoretical Computer Science*, vol. 320, no. 1, pp. 123 – 134, 2004. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0304397504001987>
- [14] C. R. Müller, J. T. V. Leuven, H. A. Wichman, and P. Joyce, "Selecting among three basic fitness landscape models: Additive, multiplicative and stickbreaking," *Theoretical Population Biology*, vol. 122, pp. 97 – 109, 2018.
- [15] E. van Nimwegen, J. P. Crutchfield, and M. Mitchell, "Statistical dynamics of the royal road genetic algorithm," *Theoretical Computer Science*, vol. 229, pp. 41–102, 1999.
- [16] P. K. Lehre, "Fitness-levels for non-elitist populations," in *Proc. of GECCO'11*, 2011, pp. 2075–2082.
- [17] D. Corus, D. Dang, A. V. Eremeev, and P. K. Lehre, "Level-Based Analysis of Genetic Algorithms and Other Search Processes," *IEEE Trans. on Evolutionary Computation*, vol. 22, no. 5, pp. 707–719, Oct. 2018.
- [18] D.-C. Dang and P. K. Lehre, "Runtime analysis of non-elitist populations: From classical optimisation to partial information," *Algorithmica*, vol. 75, no. 3, pp. 428–461, 2016.
- [19] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., 1989.
- [20] A. V. Spirov and A. V. Eremeev, "Modularity in biological evolution and evolutionary computation," *Uspekhi Sovremennoi Biologii*, vol. 139, no. 6, pp. 523–539, 2019, (in Russian). [Online]. Available: <https://doi.org/10.1134/S0042132419060073>
- [21] T. Paixao, G. Badkobeh, N. Barton, D. Corus, D.-C. Dang, T. Friedrich, P. K. Lehre, D. Sudholt, A. M. Sutton, and B. Trubenova, "Toward a unifying framework for evolutionary processes," *Journal of Theoretical Biology*, vol. 383, pp. 28 – 43, 2015.
- [22] F. Neumann, P. S. Oliveto, and C. Witt, "Theoretical analysis of fitness-proportional selection: Landscapes and efficiency," in *Proc. of GECCO'09*, 2009, pp. 835–842.
- [23] D.-C. Dang, A. Eremeev, and P. K. Lehre, "Runtime analysis of fitness-proportionate selection on linear functions," *ArXiv*, vol. 1908.08686 [cs.NE], 2019.
- [24] P. K. Lehre, "Negative drift in populations," in *Proc. of PPSN'10*, 2010, pp. 244–253.
- [25] S. Droste, T. Jansen, and I. Wegener, "On the analysis of the (1+1) evolutionary algorithm," *Theoretical Computer Science*, vol. 276, no. 1-2, pp. 51–81, Apr. 2002.
- [26] P. S. Oliveto and C. Witt, "Simplified drift analysis for proving lower bounds in evolutionary computation," *Algorithmica*, vol. 59, no. 3, pp. 369–386, 2011.
- [27] —, "On the runtime analysis of the simple genetic algorithm," *Theor. Comput. Sci.*, vol. 545, pp. 2–19, 2014.
- [28] —, "Improved time complexity analysis of the simple genetic algorithm," *Theor. Comput. Sci.*, vol. 605, pp. 21–41, 2015.
- [29] B. Doerr and T. Kotzing, "Multiplicative up-drift," in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. GECCO'19. New York, NY, USA: ACM, 2019, pp. 1470–1478.