Fitness-Levels for Non-Elitist Populations

Per Kristian Lehre*
DTU Informatics
Technical University of Denmark
Kongens Lyngby, Denmark
pkle@imm.dtu.dk

ABSTRACT

This paper introduces an easy to use technique for deriving upper bounds on the expected runtime of non-elitist population-based evolutionary algorithms (EAs). Applications of the technique show how the efficiency of EAs is critically dependant on having a sufficiently strong selective pressure. Parameter settings that ensure sufficient selective pressure on commonly considered benchmark functions are derived for the most popular selection mechanisms. Together with a recent technique for deriving lower bounds, this paper contributes to a much-needed analytical tool-box for the analysis of evolutionary algorithms with populations.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity

General Terms

Theory, Algorithms

Keywords

Runtime Analysis, Evolutionary Algorithms

1. INTRODUCTION

Despite the often very complex behaviour of evolutionary algorithms (EAs), there have been significant advances in the theoretical understanding of these algorithms in the past decade. In particular, there is now a large number of rigorously proved results on the time-complexity of EAs [13]. One contributing factor behind this success may have been the clear strategy to initiate the analysis on the simplest settings before proceeding to more complex scenarios, while at the same developing appropriate analytical techniques.

*Supported by Deutsche Forschungsgemeinschaft (DFG) under grant number WI 3552/1-1.

One of the simplifying assumptions made was to disregard the population, and focus on variants of the simple (1+1) EA which only keep one individual after each generation [3]. Runtime results quickly emerged, first for simple example problems that exhibited some fundamental structure, and later to more complex problems, including classical combinatorial optimisation problems. Progress was also made along another direction, to understand a wider range of search heuristics, including ant colony optimisation, particle swarm optimisation, and memetic algorithms. The progress was made possible by the development (and discovery) of appropriate analytical techniques tailored to runtime analysis of EAs. One of the first general techniques that was developed is artificial fitness levels [15].

When applicable, artificial fitness levels is one of the conceptually simplest ways of deriving *upper bounds* on the expected runtime of elitist EAs. (An EA is called elitist when at least one of the best individuals in the current population is copied unchanged to the next population.) The idea is to partition the search space $\mathcal{X}$ into so-called *fitness levels* $A_1, \dots, A_m \subseteq \mathcal{X}$, such that for all $i \in [m]$, all the search points in fitness level A_i have inferior function value to the search points in fitness level A_{i+1}. Due to the elitism, the EA will never loose the highest fitness level found so far. If the probability of mutating any search point in fitness level A_j into one of the higher fitness levels is at least s_j, then the expected time until this occurs is at most $1/s_j$. The expected time to overcome all the inferior levels, i. e., the expected runtime, is by linearity of expectation no more than $\sum_{j=1}^{m-1} 1/s_j$. This simple technique can sometimes provide tight upper bounds. Sudholt recently provided an alternative fitness-level argument for deriving *lower bounds* [14]. The key idea behind the technique is a condition which when satisfied ensures that the EA will not skip too many fitness levels when optimising the function. The technique is applicable to a wide range of EAs, and provides remarkably tight bounds in several cases.

Some techniques have emerged for analysing EAs with populations. Witt introduced a family tree technique for the (μ+1) EA and other EAs [16]. Chen et al. introduced a technique for deriving upper bounds on the expected runtime of *elitist* EAs which is similar in flavour to the fitness-level arguments [1]. Lässig and Sudholt used a fitness-level technique for parallel EAs [9]. Neumann et al. [12] adapted drift analysis to fitness-proportionate selection. The approach involves finding a way to represent the state of the entire population by a single real value. Based on work in [11], Lehre recently introduced an alternative drift theorem specifically for population-based EAs, which decouples the effects of the selection mechanism from the effects of the variation operator [10].

Our contribution is a fitness-level technique for obtaining upper bounds on *non-elitist* population-based EAs, thus complementing the result in [10]. The paper is organised as follows. Section 2 specifies the class of algorithms that are considered. Section 3 states the main theorem, i. e., the new fitness-level technique. Section 4 analyses the cumulative selection probability and reproduction rate in the most popular selection mechanisms. The results from this analysis are used to verify Condition 3 in the main theorem. Section 5 illustrates how to apply the fitness-level technique to obtain upper bounds on a range of pseudo-Boolean functions. Section 6 complements the previous section by providing lower bounds obtained by means of the techniques in [10]. Section 7 concludes the paper.

2. PRELIMINARIES

The j-th element of a vector P is denoted by $P(j)$. For any positive integer n, define $n := \{1, 2, \ldots, n\}$. For any predicate $\mathcal{A}$, the expression $[\mathcal{A}]$ takes the value 1 if $\mathcal{A}$ is true, and 0 otherwise. The Hamming-distance is denoted by $H(\cdot,\cdot)$. Standard notation for asymptotic growth of functions is used (see, e.g., [2]). The natural logarithm is denoted by $\ln(\cdot)$, and the logarithm to the base 2 is denoted by $\log(\cdot)$. For a bitstring x of length n, define $|x|_1 := \sum_{i=1}^{n} x_i$.

The main theorem is a statement about the runtime of the search heuristics covered by the algorithmic scheme in Algorithm 1. The algorithmic scheme defines a class of algorithms which can be instantiated by specifying the variation operator p_{mut}, and the selection mechanism p_{sel}. The neutral term population selection-variation algorithm is used to emphasise that the algorithmic scheme not only encompasses evolutionary algorithms, but also other population-based search heuristics.

1 Population Selection-Variation Algorithm

```
Require: Finite state space 𝒳,
    transition matrix p_mut over 𝒳,
    and initial population P_0 ~ Unif(X^λ).
for t = 0, 1, 2, ... until termination condition met do
    for i = 1 to λ do
        Sample I_t(i) ∈ [λ] according to p_sel(P_t).
        x := P_t(I_t(i)).
        Sample x' according to p_mut(x).
        P_{t+1}(i) := x'.
    end for
end for
```

Algorithm 1 is identical to the algorithmic scheme introduced in [10], except for one assumption about p_{sel} that will be described below. The algorithm keeps a vector $P_t \in \mathcal{X}^\lambda, t \geq 0$, of λ search points. In analogy with evolutionary algorithms, the vector will be referred to as a *population*, and the vector elements as *individuals*. Each iteration of the inner loop is called a *selection variation*-step. Then, λ iterations of the inner loop, i. e., one iteration of the outer loop, is called a *generation*. The initial population is sampled uniformly at random. In subsequent generations, a new population P_{t+1} is generated by independently sampling λ individuals from the existing population P_t according to p_{sel}, and perturbing each of the sampled individuals by a variation operator p_{mut}. Note that the generations are non-overlapping, so the algorithm is non-elitist.

Variation operators are formally represented as transition matrices $p_{\mathrm{mut}} : \mathcal{X} \times \mathcal{X} \to [0, 1]$ over the search space, where $p_{\mathrm{mut}}(x \mid y)$ represents the probability of perturbing an individual y into an individual x. Hence, the algorithmic scheme is restricted to *unary* variation operators, i. e., those where each individual has only one parent. Higher-arity variation operators (e. g., crossover operators) are not covered.

Selection operators are represented as probability distributions over the set of integers $[\lambda]$, where the conditional probability $p_{\mathrm{sel}}(i \mid P_t)$ represents the probability of selecting individual $P_t(i)$, i. e., the i-th individual from population P_t. For notational convenience, the probability of selecting an individual $x \in P_t$ will also be denoted by $p_{\mathrm{sel}}(x \mid P_t)$. Each individual within a generation is sampled independently from the same distribution p_{sel}. Note that this assumption is not made in [10]. The fitness function $f : \mathcal{X} \to \mathbb{R}$ is considered implicitly given by the selection mechanism. Without loss of generality, we assume maximisation. The ordering of the elements in a population vector $P \in \mathcal{X}^\lambda$ according to decreasing f-value will be denoted $x_{(1)}, x_{(2)}, \ldots, x_{(\lambda)}$, i. e., such that $f(x_{(1)}) \geq f(x_{(2)}) \geq \cdots \geq f(x_{(\lambda)})$. For any constant $\gamma \in (0, 1)$, the individual $x_{(\lceil \gamma\lambda \rceil)}$ will be referred to as the *γ-ranked individual* of the population.

Definition 1. *A selection mechanism p_{sel} is* f-monotone *if for all $P \in \mathcal{X}^\lambda$ and pairs $i, j \in [\lambda]$ it holds*

$$p_{sel}(i \mid P) \geq p_{sel}(j \mid P) \Longleftrightarrow f(P(i)) \geq f(P(j)).$$

Informally, the *selective pressure* of a selection mechanism refers to the degree to which the selection mechanism selects individuals that have higher f-values, and will be quantified in two different ways. The fitness-level technique (Theorem 4) for proving upper bounds uses the *cumulative selection probability*, whereas the negative drift theorem (Theorem 12) for proving lower bounds uses the *reproductive rate*.

Definition 2. *The* cumulative selection probability β *associated with selection mechanism p_{sel} is defined for all $\gamma \in (0, 1]$ and $P \in \mathcal{X}^\lambda$ by*

$$\beta(\gamma, P) := \sum_{y \in P} p_{sel}(y \mid P) \cdot \left[f(y) \geq f(x_{(\gamma\lambda)})\right]$$

Informally, $\beta(\gamma, P)$ is the probability of selecting an individual with fitness at least as high as that of the γ-ranked individual. In the cases where β is independent of the population vector P, we will write $\beta(\gamma)$ instead of $\beta(\gamma, P)$.

Definition 3. *The* reproductive rate *of Algorithm 1 is*

$$\alpha_0 := \inf_{t \geq 0} \max_{1 \leq i \leq \lambda} \mathbf{E}\left[R_t(i)\right],$$

where $R_t(i) := \sum_{j=1}^{\lambda} [I_t(j) = i]$.

Informally, the reproductive rate is the expected number of times the individual with highest selection probability is selected per generation. The reproductive rate is always defined, and in the interval $[1, \lambda]$. The reproductive rate α_0 sets a limit on the expected number of offspring per individual in the population. A reproductive rate close to 1 indicates a low selective pressure.

3. MAIN RESULT

Recall the definition of an f-based partition [15]: A *partition* of a finite set $\mathcal{X}$ is a collection of subsets $A_1, \dots, A_{m+1}$ such that $A_i \cap A_j = \emptyset$ for all $i \neq j$, and $A_1 \cup \dots \cup A_{m+1} = \mathcal{X}$. A partition is called *f-based* if for all pairs $x \in A_i$ and $y \in A_j$, it holds that $f(x) \leq f(y)$ if and only if $i \leq j$, and A_{m+1} contains only optimal search points. For notational convenience, define for $j \in [m]$, the set $A_j^+ := \bigcup_{i=j+1}^{m+1} A_i$, i.e., the set of search points at higher fitness levels than A_j.

THEOREM 4. *Given a function $f : \mathcal{X} \to \mathbb{R}$, and an f-based partition $(A_1, \dots, A_{m+1})$, let T be the number of function evaluations until Algorithm 1 with an f-monotone selection mechanism p_{sel} obtains an element in A_{m+1} for the first time. If there exists $p_0, s_1, \dots, s_m, s_* \in (0,1]$, and constants $\gamma_0, \varepsilon \in (0,1)$, and $\delta > 0$, such that for all $P \in \mathcal{X}^\lambda$, for all $\gamma \in (0, \gamma_0)$, and for all $j \in [m]$, it holds*

(C1) $p_{mut}\left(y \in A_j^+ \mid x \in A_j\right) \geq s_j \geq s_*$

(C2) $p_{mut}\left(y \in A_j \cup A_j^+ \mid x \in A_j\right) \geq p_0$

(C3) $\dfrac{\beta(\gamma, P)}{\gamma} \geq \dfrac{(1+\delta)}{p_0}$

(C4) $\lambda \geq \dfrac{2(1+\delta)}{\varepsilon\delta^2} \cdot \ln\left(\dfrac{m}{s_*}\right)$

Then $\mathbf{E}[T] \leq c(m\lambda^2 + \sum_{j=1}^m s_j^{-1})$ *for some constant $c > 0$.*

Before proving the theorem, we informally describe the conditions of the theorem. Similarly to the classical fitness-level argument, the theorem assumes that an f-based partition is provided. Four conditions must be satisfied for the theorem to hold:

The first condition requires that for each fitness level j, there is a lower limit s_j on the "upgrade probability" from level j, i.e., the probability of mutating an individual x in fitness level j into an individual y that belongs to a higher fitness level. This condition is the same as for the classical fitness level argument discussed in the introduction.

The second condition requires that there exists a lower limit p_0 on the probability that the individual will not "downgrade" to a lower fitness level. In the classical setting of bitwise mutation with mutation rate $1/n$, it suffices to use any parameter $p_0 \leq (1/e)(1-1/n)$, which is less than the probability of not mutating any bits.

The third condition requires that the selective pressure induced by the selection mechanism is sufficiently strong. The selective pressure is quantified via the cumulative selection probability β (see Definition 2). The required level of selective pressure depends on parameter p_0 in the second condition.

The fourth condition requires that the population size λ is sufficiently large. The required population size depends on the number of fitness levels m and the upgrade probabilities s_j, which are problem-dependant parameters. However, a population size of $\lambda = \Theta(\log n)$, where n is the problem dimension, will suffice for many pseudo-Boolean functions.

If the four conditions can be satisfied, then an upper bound on the expected runtime of the algorithm can be guaranteed. The upper bound depends on the number of fitness levels m, on the population size λ, and on the upgrade probabilities s_j, for $j \in [m]$.

PROOF OF THEOREM 4. We bound the expected runtime by estimating the expected number of generations until the γ-ranked individual reaches the optimal fitness level A_{m+1}. Let τ denote the number of generations until this occurs. Let $\mathcal{E}$ be the event that if the γ-ranked individual in the current generation belongs to fitness level $A_j, j \in [m]$, then the γ-ranked individual in the next generation belongs to $A_j \cup A_j^+$.

We divide the run of the algorithm into phases, each phase lasting for $2t_\mathcal{E}$ generations, where $t_\mathcal{E} := \mathbf{E}[\tau \mid \mathcal{E}]$, i.e., $t_\mathcal{E}$ is the expected runtime in generations given that event $\mathcal{E}$ always holds. Assuming that event $\mathcal{E}$ always holds, the γ-ranked individual reaches the optimal fitness level within one phase with probability at least $1/2$ by Markov's inequality. A phase is considered *successful* if event $\mathcal{E}$ holds during the phase, and the γ-ranked individual has reached the optimal fitness level before the end of the phase. A phase is therefore successful with probability at least $p_\mathcal{E}/2$, where $p_\mathcal{E}$ is the probability that event $\mathcal{E}$ holds during the phase. The expected number of phases before a successful phase is no more than $2/p_\varepsilon$. It follows that the expected, unconditional runtime is $\mathbf{E}[T] \leq 4\lambda t_\mathcal{E}/p_\mathcal{E}$. In the remaining of the proof, we estimate $t_\mathcal{E}$, and show that $p_\mathcal{E}$ is bounded from below by a positive constant.

We begin by finding an upper bound for $t_\mathcal{E}$. For all $j \in [m]$, define the random variable τ_j to be the number of generations, starting with the γ-ranked individual in fitness level A_j, until the γ-ranked individual reaches a strictly better fitness level A_k for $k > j$. Linearity of expectation gives $t_\mathcal{E} = \mathbf{E}[\tau \mid \mathcal{E}] \leq \sum_{j=1}^m \mathbf{E}[\tau_j \mid \mathcal{E}]$, so we can focus on bounding the conditional expectation of τ_j for any j. Assume that the γ-ranked individual belongs to fitness level A_j in generation t. We call an individual *advanced* if it belongs to A_j^+, i.e. any of the strictly better fitness level than the γ-ranked individual. For all $t \geq 0$, let random variable X_t be defined as the number of advanced individuals in generation t.

We now make some observations about the random variable X_{t+1}. We can consider each selection and mutation step in a generation as an independent trial, where a trial is successful if the outcome of the selection and mutation step is an advanced individual. Random variable X_{t+1} is therefore binomially distributed, corresponding to the number of successes in λ trials. Assuming that the previous generation contained at least one advanced individual, the expectation of X_{t+1} can be bounded using Condition 3 by

$$\mu_{t+1} := \mathbf{E}[X_{t+1} \mid 1 \leq X_t < \lceil\gamma\lambda\rceil] \tag{1}$$

$$\geq \lambda \cdot \beta(X_t/\lambda) \cdot p_0 \geq X_t \cdot (1+\delta). \tag{2}$$

The probability of a successful trial can be significantly different in the case when $X_t = 0$, i.e., when there are no advanced individuals. Define q_j as the probability that the next generation contains at least one advanced individual. This probability can be bounded as

$$q_j := \Pr(X_{t+1} > 0 \mid X_t = 0) > 1 - (1 - \beta(\gamma)s_j)^\lambda$$

$$\geq 1 - \exp(-\lambda\beta(\gamma)s_j) \geq \frac{\lambda\beta(\gamma)s_j}{\lambda\beta(\gamma)s_j + 1},$$

where the last inequality follows from Lemma 17.

The random variable τ_j can be expressed as $\tau_j = \min\{t \geq 0 \mid X_t \geq \lceil\gamma\lambda\rceil\}$. To bound the expectation of τ_j, we can therefore apply Theorem 16 (the drift theorem) with respect to the stochastic process X_t. We will use the distance function $V(x) := g(x) + h(x)$ with the two components $g(x) := \lceil\gamma\lambda\rceil - x$, and $h(x) := 1/q_j e^{\kappa x}$, where κ can be chosen arbitrarily such that $\kappa \in (0, \delta)$. The maximal distance is bounded from above by $B := \lambda + 1/q_j$.

We will show that there exists a constant $\Delta > 0$ such that

$$\mathbf{E}\left[V(X_t) - V(X_{t+1}) \mid \mathcal{E}, X_t = i\right] = \mathbf{E}\left[\Delta_{g,t}(i) + \Delta_{h,t}(i)\right] \geq \Delta$$

for all i, where the two drift components are defined as

$$\Delta_{g,t}(i) := (g(X_t) - g(X_{t+1}) \mid \mathcal{E}, X_t = i), \text{ and}$$
$$\Delta_{h,t}(i) := (h(X_t) - h(X_{t+1}) \mid \mathcal{E}, X_t = i).$$

Case 1 ($i \geq 1$): Eq. (2) gives

$$\mathbf{E}\left[\Delta_{g,t}(i) \mid i \geq 1\right] = \mathbf{E}\left[X_{t+1} - i \mid \mathcal{E}, X_t = i \geq 1\right] \geq \delta i.$$

Furthermore, since X_{t+1} conditional on $X_t = i \geq 1$ is binomially distributed with parameters λ and $p \geq (1+\delta)i/\lambda$, it follows from Lemma 18 that

$$\mathbf{E}\left[\Delta_{h,t}(i) \mid i \geq 1\right] = \frac{1}{q_j} \cdot \mathbf{E}\left[e^{-i\kappa} - e^{-\kappa X_{t+1}} \mid \mathcal{E}, i \geq 1\right] \geq 0.$$

Case 2 ($i = 0$): It suffices to use the trivial lower bound $\mathbf{E}\left[\Delta_{g,t}(i) \mid i = 0\right] \geq 0$ for the first drift component. The next generation contains at least one advanced individual with probability q_j. So for the second drift component, it holds that

$$\mathbf{E}\left[\Delta_{h,t}(i) \mid i = 0\right] \geq q_j \cdot (1/q_j) \cdot (e^0 - e^{-\kappa}) = 1 - e^{-\kappa}.$$

By linearity of expectation, it therefore holds for all $i \geq 0$ that $\mathbf{E}\left[\Delta_{g,t}(i) + \Delta_{h,t}(i)\right] \geq \min\{\delta, 1 - e^{-\kappa}\} =: \Delta$. Theorem 16 now gives

$$\mathbf{E}\left[\tau_j \mid \mathcal{E}\right] \leq \frac{B}{\Delta} = \frac{1}{\Delta}\left(\lambda + \frac{1}{q_j}\right) \leq \frac{1}{\Delta}\left(\lambda + \frac{1}{\lambda s_j \beta(\gamma)} + 1\right).$$

Summing up for all fitness levels gives

$$t_{\mathcal{E}} = \mathbf{E}\left[\tau \mid \mathcal{E}\right] \leq \frac{1}{\Delta}\left(m(\lambda+1) + \sum_{j=1}^{m} \frac{1}{\lambda s_j \beta(\gamma)}\right).$$

We then estimate the probability $p_{\mathcal{E}}$. To produce an individual that is as at least as fit as the γ-ranked individual, it suffices to select any of the best $\lceil \gamma\lambda \rceil$ individuals in the population and not create an offspring that belongs to a lower fitness level. Using Condition 3, the probability of this event is at least $\beta(\gamma)p_0 > (1+\delta)\gamma$. Hence, the expected number of times this happens during one generation is at least $(1+\delta)\lceil \gamma\lambda \rceil$. In order to reduce the fitness of the γ-ranked individual, it is necessary that there are less than $\lceil \gamma\lambda \rceil$ such events. However, by a Chernoff bound, the probability r that this happens in a given generation is

$$r \leq \exp\left(-\frac{\varepsilon\lambda\delta^2}{2(1+\delta)}\right) \exp\left(-\frac{(1-\varepsilon)\lambda\delta^2}{2(1+\delta)}\right) \leq \frac{s_*}{m} e^{-\varepsilon'\lambda}$$

where the last inequality follows from condition 4, and we defined the constant $\varepsilon' := \delta^2(1-\varepsilon)/2(1+\delta)$. The probability $p_{\mathcal{E}}$ that event $\mathcal{E}$ holds during $2t_{\mathcal{E}}$ generations is by union bound $p_{\mathcal{E}} \geq 1 - 2t_{\mathcal{E}} r$, so

$$p_{\mathcal{E}} \geq 1 - \frac{2m}{\Delta}\left(\lambda + 1 + \frac{1}{\lambda s_* \beta(\gamma)}\right) \frac{s_*}{m} e^{-\varepsilon'\lambda} = 1 - e^{-\Omega(\lambda)},$$

where the asymptotic expression is with respect to λ.

The proof is completed by choosing a sufficiently large constant $c > 0$ such that $4\lambda t_{\mathcal{E}}/p_{\mathcal{E}} \leq c(m\lambda^2 + \sum_{j=1}^{m} s_j^{-1})$. □

4. MEASURING SELECTIVE PRESSURE

Section 5 illustrates how Theorem 4 can be applied to derive upper bounds on the expected runtime of evolutionary algorithms with some classical selection mechanisms. This section focuses on Condition 3 of Theorem 4 only, showing that it often can be verified independently of the function f that is optimised. We analyse how the parameter settings of the most popular selection mechanisms influence the selective pressure, as measured by the cumulative selection probability, and the reproductive rate. In particular, we will determine the parameter regions for which Condition 3 of Theorem 4 is satisfied (called parameter region of "high" selective pressure), and the parameter regions for which Condition 2 of Theorem 16 is satisfied (called parameter region of "low" selective pressure). The statements are given in Lemmas 5, 6, 7, and 8, and in Corollary 10. The results are summarised in Table 1 for the case where p_{mut} is the standard bitwise mutation with probability χ/n. In this case, any constant $p_0 < 1/e^{\chi}$ can be used independently of the problem.

Note the following about the results in Table 1. For all selection mechanisms except fitness proportional selection, there is a sharp transition between the parameter region where the selective pressure is sufficiently high, and the parameter region where the selective pressure is too low. Furthermore, the transition between the two parameter regions can depend in a non-linear way on other parameters, notably the mutation rate parameter. Standard parameter settings for mutation rate ($\chi = 1$) and the selection mechanisms sometimes put the algorithm in the low selective pressure region. In particular, linear ranking selection with standard mutation rate $1/n$ is always in the low selection pressure region, because $1 \leq \eta < 2$. Hence, in order to put linear ranking selection in the high selective pressure region, it is necessary to decrease the mutation rate χ below $\ln 2 \approx 0.69$. Also, unscaled fitness proportionate selection ($\nu = 1$) has insufficient selective pressure on OneMax. Finally, binary tournament selection ($k = 2$) has insufficient selective pressure with mutation rate $1/n$. To enter the high selective pressure region, it is necessary to increase the tournament size to at least 3, or to decrease the mutation rate.

Tournament Selection

In tournament selection with tournament size $k \geq 2$, one selects the fittest individual among k uniformly sampled individuals with replacement from the current population. Ties are broken uniformly at random.

Lemma 5. *The reproductive rate of k-Tournament Selection is at most k. If $k \geq (1+\delta)/p_0$ for any constants $p_0 \in (0,1)$ and $\delta > 0$, then there exist constants $\gamma, \delta' > 0$ such that $p_0\beta(\gamma') > \gamma'(1+\delta')$ for all $\gamma' \in (0, \gamma]$.*

Proof. For the first statement, the probability that a given individual occurs in a tournament is by union bound at most k/λ. Hence, the expected number of times a given individual is selected is at most $\lambda \cdot (k/\lambda) = k$.

For the second statement, in order to select an individual with the same fitness or better fitness than the γ-ranked individual, it is sufficient that the randomly sampled tournament contains at least one individual with rank γ or better. Hence, one obtains for $0 < \gamma < 1$,

$$\beta(\gamma) > 1 - (1-\gamma)^k. \tag{3}$$

Selection Mech.	High S. P.	Low S. P.	Problem	Pop. Size	High S. P.	Low S. P.
Fitness Prop. (*)	$\nu > f_{\max}\ln(2e^{\chi})$	$\nu < \chi/\ln 2, n^3 \leq \lambda$	OneMax	$\lambda \geq c\ln n$	$O(n\lambda^2)$	$e^{\Omega(n)}$
Linear Ranking	$\eta > e^{\chi}$	$\eta < e^{\chi}$	LeadingOnes	$\lambda \geq c\ln n$	$O(n\lambda^2 + n^2)$	$e^{\Omega(n)}$ [10]
k-Tournament	$k > e^{\chi}$	$k < e^{\chi}$	Linear Functions	$\lambda \geq c\ln n$	$O(n\lambda^2 + n^2)$	$e^{\Omega(n)}$ [10]
(μ, λ)	$\lambda > \mu e^{\chi}$	$\lambda < \mu e^{\chi}$	k-Unimodal	$\lambda \geq c\ln(nk)$	$O(k\lambda^2 + nk)$	$e^{\Omega(n)}$ [10]
Cellular EAs		$\Delta(G) < e^{\chi}$ [10]	Jump$_r$	$\lambda \geq cr\ln n$	$O(n\lambda^2 + (n/\chi)^r)$	$e^{\Omega(n)}$ [10]

Table 1: Left: Separation of the parameter spaces of five selection mechanisms into regions of "high" and "low" selective pressure (S. P.) (cf. Lemma 5-8, and Corollary 10), assuming bitwise mutation rate χ/n. For notational clarity, $(1 \pm \delta)$-factors are omitted. Right: Expected runtime of Algorithm 1 with corresponding parameter settings (cf. Theorems 11 and 14). (*) The result for fitness prop. selection is only for OneMax.

From the assumption that $k \geq (1+\delta)/p_0$, it follows that

$$(1-\gamma)^k < e^{-\gamma k} < \frac{1}{1+\gamma k} \leq \frac{p_0}{p_0 + \gamma(1+\delta)}. \quad (4)$$

Choose γ sufficiently small such that $\gamma < \delta p_0/2(1+\delta)$ holds. Combining (3) and (4), then yields the desired result

$$\beta(\gamma) > \frac{\gamma(1+\delta)}{p_0 + \gamma(1+\delta)} > \frac{\gamma(1+\delta)}{p_0(1+\delta/2)}. \quad \square$$

Linear Ranking Selection

Ranking selection mechanisms select individuals according to the fitness rank in the population. Individuals are ranked from 0 to 1, with the best individual ranked 0, and the worst individual ranked 1. Following Goldberg and Deb [4], a function $\alpha : \mathbb{R} \to \mathbb{R}$ is considered a ranking function if $\alpha(x) \geq 0$ for all $x \in [0, 1]$, and $\int_0^1 \alpha(y)dy = 1$. The selection mechanism is defined by the cumulative probability of selecting individuals ranked γ or better, by $\int_0^{\gamma} \alpha(x)dx$. Linear ranking selection is defined by setting $\alpha(\gamma) := \eta(1-2\gamma) + 2\gamma$, where the parameter $\eta \in (1, 2]$ adjusts the selective pressure.

Lemma 6. *The reproductive rate of linear ranking selection is not larger than η. Furthermore, if $\eta > (1+\delta)/p_0$ for some $\delta > 0$, then there exists a constant $\gamma > 0$ such that $p_0\beta(\gamma') > \gamma'(1+\delta/2)$ for any $\gamma' \in (0, \gamma)$.*

Proof. The expected number of times a given individual is selected is bounded from above by $\alpha(0) = \eta$ (see [11]), so the statement about the reproductive rate holds.

If $\eta > (1+\delta)/p_0$ and γ is chosen sufficiently small so that $(1+\delta)(1-\gamma) > (1+\delta/2)$, then

$$p_0\beta(\gamma) \geq p_0 \int_0^{\gamma} \alpha(x)dx = p_0\gamma(\eta(1-\gamma)+\gamma) > \gamma(1+\frac{\delta}{2}). \quad \square$$

(μ, λ)-Selection

In (μ, λ)-selection, one selects uniformly at random among the best μ out of λ individuals in the current population.

Lemma 7. *The reproductive rate of (μ, λ)-selection is no more than λ/μ. If $\lambda/\mu > (1+\delta)/p_0$ for some constant $\delta > 0$, then $p_0\beta(\gamma) \geq \gamma(1+\delta)$ for all $\gamma \in (0, \mu/\lambda)$.*

Proof. The probability of selecting a given individual is no more than $1/\mu$, so the reproductive rate is no more than λ/μ. Furthermore, $\beta(\gamma) \geq \gamma\lambda/\mu$ if $\gamma\lambda \leq \mu$, and $\beta(\gamma) = 1$ otherwise. Hence, if $\lambda/\mu > (1+\delta)/p_0$ for some constant $\delta > 0$, then $\beta(\gamma) \geq \gamma\lambda/\mu > \gamma(1+\delta)/p_0$ for any $\gamma < \mu/\lambda$. $\square$

Fitness Proportionate Selection

Fitness proportionate selection with power scaling parameter $\nu \geq 1$ is defined for maximisation problems as follows

$$\forall i \in [\lambda] \quad p_{\text{sel}}(i \mid P_t, f) := \frac{f(P_t(i))^{\nu}}{\sum_{j=1}^{\lambda} f(P_t(j))^{\nu}}.$$

Setting $\nu = 1$ gives classical fitness proportionate selection.

Happ et al. considered variants of the RLS and the (1+1) EA where the plus-selection mechanism is replaced with fitness proportionate selection [5]. They found that changing the algorithms this way makes them highly inefficient on linear functions. However, their analysis is limited to EAs with a population size of one, which is uncommon in applications of fitness proportionate selection. Neumann et al. showed that, unless the fitness function is scaled, even larger population sizes are unhelpful on OneMax [12]. In the following, it is shown how similar results can be obtained using the techniques presented here and in [10].

Lemma 8. *Let $f_{\max}$ be any integer. Fitness proportionate selection with scaling parameter $\nu > \ln(2/p_0)f_{\max}$ satisfies $p_0\beta(\gamma) > \gamma(1+1/3)$ for all $\gamma < p_0/4$, on any integer-valued fitness function with range in $[0, f_{\max}]$.*

Proof. Choose any constant $\gamma < p_0/4$, and let $f_{\gamma} \leq f_{\max}$ be the fitness of the γ-ranked individual in the population. Let $k \geq \lceil \gamma\lambda \rceil$ be the number of individuals with fitness at least f_{γ}, and $s \geq kf_{\gamma}^{\nu} \geq \lceil \gamma\lambda \rceil f_{\gamma}^{\nu}$ be the scaled sum of the fitness values of these k individuals. The probability of selecting one of the k individuals is

$$\beta(\gamma) \geq \frac{s}{(\lambda-k)(f_{\gamma}-1)^{\nu}+s} \geq \frac{\gamma}{(1-k/\lambda)(1-f_{\gamma}^{-1})^{\nu}+\gamma}$$
$$\geq \frac{\gamma}{(1-f_{\max}^{-1})^{\nu}+p_0/4} \geq \frac{\gamma}{p_0/2+p_0/4},$$

and the result follows for $\delta = 1/3$. $\square$

To derive a good upper bound on the reproductive rate of fitness proportionate selection, it is necessary to bound the sum of the fitness values in the population. For functions like OneMax, it seems intuitive that the population should contain relatively few 0-bits, as the selection-mechanism has a bias towards individuals with 1-bits. The following lemma shows that this intuition is correct for fitness proportionate selection, and many other selection mechanisms.

Lemma 9. *Let $\varepsilon > 0$ be any constant. Define T to be the smallest t such that Algorithm 1 using an f-monotone selection mechanism, and population size $\lambda \geq n^3$ has a population P_t where $\sum_{j=1}^{\lambda} |P_t(j)|_1 \leq \lambda(n/2)(1-\varepsilon)$. Then $\Pr(T \leq e^{cn}) = e^{-\Omega(n)}$.*

PROOF. For the initial population, it follows by a Chernoff bound that $\Pr(T=1)=e^{-\Omega(n)}$. We then claim that for all $t \geq 0$, $\Pr(T=t+1 \mid T>t) \leq e^{-c'n}$ for a constant $c'>0$, which by the union bound implies that $\Pr(T<e^{cn}) \leq e^{cn-c'n}=e^{-\Omega(n)}$ for any constant $c<c'$.

To see why the claim holds, let $Z_t^{(j)}=n-|P_t(j)|_1$, for $t \geq 0$ and $j \in [\lambda]$, be the number of 0-bits in the j-th individual in generation t, and p_j the probability of selecting the j-th individual when producing the population in generation $t+1$. Furthermore, let $Z_t=\sum_{j=1}^{\lambda} Z_t^{(j)}$. For f-monotone selection mechanisms, it holds that $\sum_{j=1}^{\lambda} p_j Z_t^{(j)} \leq Z_t/\lambda$. The expected number of 0-bits in an offspring $j \in [\lambda]$, is

$$\begin{aligned}
&\mathbf{E}\left[Z_{t+1}^{(j)} \mid Z_t^{(k)}=z_k, 1 \leq k \leq \lambda\right] \\
&= \sum_{k=1}^{\lambda} p_k \left((n-z_k)\frac{\chi}{n}+z_k\left(1-\frac{\chi}{n}\right)\right) \\
&= \chi+\left(1-\frac{2\chi}{n}\right)\sum_{k=1}^{\lambda} p_k z_k \leq \chi+\left(1-\frac{2\chi}{n}\right)(Z_t/\lambda),
\end{aligned}$$

and the expected number of 0-bits in generation $t+1$ is

$$\begin{aligned}
&\mathbf{E}\left[Z_{t+1} \mid Z_t=z, z<\lambda(n/2)(1+\varepsilon)\right] \\
&\qquad \leq \lambda\chi+z(1-2\chi/n) \leq z-\varepsilon\lambda\chi.
\end{aligned}$$

The random variables $Z_{t+1}^{(1)}, Z_{t+1}^{(2)}, \ldots, Z_{t+1}^{(\lambda)}$ are non-negative independent random variables, each bounded from above by n. The conditions of Theorem 15 (Hoeffding's inequality) are satisfied, leading to the desired bound

$$\begin{aligned}
\Pr(Z_{t+1}>Z_t) &\leq \Pr(Z_{t+1}-\mathbf{E}[Z_{t+1}]>\varepsilon\lambda\chi) \\
&\leq \exp\left(-\frac{2(\varepsilon\lambda\chi)^2}{\lambda n^2}\right)=e^{-\Omega(n)}. \quad \square
\end{aligned}$$

It is easy to see that fitness proportionate selection is f-monotone for all $\nu \geq 1$. Hence, the following statement follows directly from Lemma 9.

COROLLARY 10. *For any constant $\delta>0$, the reproductive rate of fitness proportionate selection on* ONEMAX *when $\lambda \geq n^3$ is no more than $2^\nu+\delta$.*

PROOF. By Lemma 9, and analogously to Eq. (6), the probability of selecting the fittest individual is no more than $n^\nu/(\lambda(n/2)(1-\varepsilon))^\nu$. $\square$

5. UPPER BOUNDS

Given the results obtained in the previous section, we are now ready to illustrate how Theorem 4 can be applied to obtain upper bounds on the expected runtime of Algorithm 1. We consider the most popular selection mechanisms, and the classical example functions that are used in runtime analysis of EAs. In particular, we analyse the runtime on linear functions (including ONEMAX), unimodal functions (including LEADINGONES), and the JUMP_k function.

A pseudo-Boolean function f is called *linear* if there exists constants $c_1,\ldots,c_n$ such that $f(x)=\sum_{j=1}^n c_i x_i$. We assume that $c_1 \geq c_2 \geq \cdots c_n > 0$. The special case where $c_1=\cdots=c_n=1$ is called ONEMAX.

The JUMP function is defined as ONEMAX, except that the optimum is separated from other search points by a Hamming-gap of inferior search points [3].

$$\text{JUMP}_r(x) := \begin{cases} |x|_1+1 & \text{if } |x|_1 \leq n-r \text{ or } |x|_1=n, \\ 0 & \text{otherwise.} \end{cases}$$

A pseudo-Boolean function f is called *unimodal* if every bitstring x is either optimal, or has a Hamming-neighbour x' such that $f(x')>f(x)$. We say that a unimodal function is k-unimodal if it has k distinct function values $f_1<f_2<\cdots<f_k$. For the lower bounds presented in Table 1, we assume that the number of search points with the optimal function value f_k is bounded from above by a polynomial in the problem size n. Note that the function $\text{LEADINGONES}(x):=\sum_{i=1}^n \prod_{j=1}^i x_i$ is n-unimodal.

The fitness partitions that are used in the proof of Theorem 11 are similar to those employed in previous applications of the fitness-level technique [8, 9, 14].

THEOREM 11. *Algorithm 1 with bit-wise mutation rate χ/n for any constant $\chi>0$, and where p_{sel} is either linear ranking selection, k-tournament selection, or (μ,λ)-selection where the parameter settings satisfy "High S. P." in Table 1 (left) and "Pop. Size" in Table 1 (right), has expected runtimes as indicated by "High S. P." in Table 1 (right). Algorithm 1 has expected runtime $O(n\lambda^2)$ on* ONEMAX *when p_{sel} is fitness prop. selection with parameter $\nu>n\ln(2e^\chi)$.*

PROOF. We apply Theorem 4 with the following f-based partitions and upgrade probabilities.

For linear functions f, we set $m=n$ and choose, as in [8],

$$A_j := \left\{x \in \{0,1\}^n \mid \sum_{i=1}^{j} c_i \leq f(x) < \sum_{i=1}^{j+1} c_i\right\}.$$

For ONEMAX, LEADINGONES, and k-unimodal functions, with k distinct function values $f_1<\cdots<f_k$, we set $m=k$ and use the canonical partition [14]

$$A_j := \{x \in \{0,1\}^n \mid f(x)=f_j\}.$$

For linear functions, and for k-unimodal functions (including LEADINGONES where $k=n$) it is sufficient to flip one specific bit, and no other bits to reach a higher fitness level. For these functions, we therefore choose for all j the upgrade probabilities $s_j:=s_*:=(\chi/n)(1-\chi/n)^{n-1}=\Omega(1/n)$.

For ONEMAX, it is sufficient to flip one of the $n-j$ 0-bits, and no other bits to escape fitness level j. For this function, we therefore choose the upgrade probabilities

$$s_j := (n-j)(\chi/n)(1-\chi/n)^{n-1}=\Omega(1-j/n), s_*:=s_{n-1}.$$

For JUMP_r, we set $m:=n-r+1$, and choose the partitions

$$\begin{aligned}
& A_1=\{x \in \{0,1\}^n \mid n-m \leq |x|_1 < n\}, \\
j>1 \quad & A_j=\{x \in \{0,1\}^n \mid |x|_1=j\}.
\end{aligned}$$

In order to escape fitness level A_1, it is sufficient to flip at most r bits, and to flip no other bits. For fitness level $j>1$, it suffices to flip one of $n-j$ 0-bits, and no other bits. Hence, we choose the upgrade probabilities

$$\begin{aligned}
& s_1 := s_* := (\chi/n)^r(1-\chi/n)^{n-r}=\Omega((\chi/n)^r) \\
j>1 \quad & s_j := (n-j)(\chi/n)(1-\chi/n)^{n-1}=\Omega(1-j/n).
\end{aligned}$$

(C1) Condition 1 is satisfied by the definition of the fitness partitions and the upgrade probabilities.

(C2) For all functions, we set the parameter p_0 to the probability of not flipping any bits. If the selected individual

x belongs to fitness level A_j, and the bitwise mutation operator does not flip any bits, then the new individual x' also belongs to fitness level A_j. Condition 2 is therefore satisfied. For any constant $\varepsilon > 0$, it holds for all $n > 2\chi^2/\ln(1+\varepsilon)$ that

$$p_0 = \left(1 - \frac{\chi}{n}\right)^n \geq \left[\left(1 - \frac{\chi}{n}\right)^{\frac{n}{\chi}-1}\right]^{\chi + \frac{2\chi^2}{n}} \geq \frac{1}{(1+\varepsilon)e^{\chi}}. \quad (5)$$

(C3) For k-tournament selection, assume for any constant $c' > 0$ that $k \geq (1+c')e^{\chi}$, and pick arbitrary positive constants ε and δ such that $1 + c' = (1+\varepsilon)(1+\delta)$. Eq. (5) implies that $k \geq (1+\varepsilon)(1+\delta)e^{\chi} \geq (1+\delta)/p_0$. Condition 3 is now satisfied by Lemma 5. Analogous arguments and Lemmas 6, 7, and 8 can be used to show that Condition 3 also holds for fitness proportionate selection, linear ranking selection, and (μ,λ)-selection.

(C4) For linear functions (including OneMax), $m/s_* = O(n^2)$. For k-unimodal functions (including LeadingOnes), $m/s_* = O(nk)$. Finally, for Jump$_r$, $m/s_* = O(n^{r+1}/\chi^r)$. So, Condition 4 is satisfied if the population size λ is set according to Table 1 (right) for a large enough constant c.

All conditions are satisfied, and the upper bounds in column "High S. P." in Table 1 (right) follow. □

6. LOWER BOUNDS

The upper bounds are complemented by lower bounds which hold for the parameter settings in the column "Low S. P." in Table 1 (left). The lower bounds are proved using Theorem 12, which is a special case of the negative drift theorem for populations (Theorem 1 in [10]). Theorem 12 is restricted to the bitwise mutation operator, but has easier to verify conditions than the general theorem.

Theorem 12 ([10]). *Given Algorithm 1 on $\mathcal{X} = \{0,1\}^n$ with bit-wise mutation rate χ/n, and population size $\lambda = \text{poly}(n)$. Let $a(n)$ and $b(n)$ be positive integers s.t. $b(n) \leq n/\chi$ and $d(n) := b(n) - a(n) = \omega(\ln n)$. For an $x^* \in \{0,1\}^n$, let $T(n)$ be the smallest $t \geq 0$, s.t. $H(P_t(j), x^*) \leq a(n)$ for some $j \in [\lambda]$. Let $R_t(i) := \sum_{j=1}^{\lambda}[I_t(j) = i]$. If there are constants $\alpha_0 \geq 1$ and $\delta > 0$ s.t. for all $t \geq 0$*

(1) $\mathbf{E}\left[R_t(i) \mid a(n) < H(P_t(i), x^*) < b(n)\right] \leq \alpha_0,\ \forall i \in [\lambda]$,

(2) $\psi := \ln(\alpha_0)/\chi + \delta < 1$, *and*

(3) $b(n)/n < \min\{\,1/5\,,\ 1/2 - 1/2\sqrt{\psi(2-\psi)}\,\}$,

then $\Pr\left(T(n) \leq e^{cd(n)}\right) = e^{-\Omega(d(n))}$ *for a constant $c > 0$.*

The theorem implies that Algorithm 1 with bitwise mutation rate χ/n has exponential runtime when the reproductive rate is below e^{χ}. The results in Table 1 for linear ranking selection, tournament selection, (μ,λ)-selection, and cellular EAs were already proved in [10]. It remains to consider fitness proportionate selection. It has been previously shown that this selection mechanism is inefficient on OneMax when scaling is not used ($\nu = 1$) [12]. The following simple corollary to Theorem 16 strengthens this result.

Corollary 13. *If $\chi = 1$, and $n^3 \leq \lambda = \text{poly}(n)$, then the probability that Algorithm 1 using fitness proportional selection with scaling parameter $\nu = 1$ obtains a search point with more than* 97.1% *1-bits when optimising* OneMax *within e^{cn} generations is $e^{-\Omega(n)}$, for some constant $c > 0$.*

Proof. We apply Theorem 12 for the parameters $a(n) = 0.02900$, $b(n) = 0.02901$, and $\alpha_0 = \frac{1-a(n)}{0.49999}$. It follows by Lemma 9 that $\sum_{j=1}^{\lambda}|P_t(j)|_1 \geq 0.49999 \cdot n\lambda$ with probability $1 - e^{-\Omega(n)}$, otherwise we consider the run a failure. The reproductive rate is therefore bounded by

$$\lambda \cdot p_{\text{sel}}(i \mid P_t, f) \leq \frac{(1-a(n)) \cdot n\lambda}{0.49999 \cdot n\lambda} = \alpha_0.$$

It is now straightforward to verify numerically that the second and third conditions are satisfied. □

Finally, it is shown that fitness proportionate selection is inefficient on OneMax, even with some degree of scaling.

Theorem 14. *Let T be the number of generations until Algorithm 1 with bitwise mutation rate χ/n, population size $n^3 \leq \lambda = \text{poly}(n)$, and fitness proportional selection with parameter $\nu, 1 \leq \nu < \frac{\chi(1-\delta)}{\ln(2+\epsilon)}$ optimises* OneMax. *Then* $\Pr\left(T \leq e^{c'n}\right) = e^{-\Omega(n)}$ *for a constant $c' > 0$.*

Proof. We apply Theorem 12, and show that Condition 1 holds by bounding the reproductive rate. We assume that $\sum_{j=1}^{\lambda}|P_t(j)|_1 \geq \lambda(n/2)(1-\varepsilon')$ holds within the first e^{cn} generations for appropriate constants $c, \varepsilon' > 0$, and consider the run a failure otherwise. By Lemma 9, the failure probability is bounded from above by $e^{-\Omega(n)}$.

The function $f(x) = x^{\nu}$ is convex for $\nu \geq 1$, so the finite form of Jensen's inequality implies that

$$\frac{1}{\lambda}\sum_{j=1}^{\lambda}|P_t(j)|_1^{\nu} \geq \left(\sum_{j=1}^{\lambda}\frac{|P_t(j)|_1}{\lambda}\right)^{\nu} \geq \left(\frac{n(1-\varepsilon')}{2}\right)^{\nu}. \quad (6)$$

Hence, the reproductive rate is bounded by

$$\max_i \frac{\lambda|P_t(i)|_1^{\nu}}{\sum_{j=1}^{\lambda}|P_t(j)|_1^{\nu}} \leq \left(\frac{n}{(n/2)(1-\varepsilon')}\right)^{\nu} = 2^{\nu}(1+\varepsilon), \quad (7)$$

for an appropriate constant ε. The second condition now follows, as $\ln(\alpha_0)/\chi < \nu\ln(2+\varepsilon)/\chi < 1-\delta$. Since $\psi < 1$, there exists a $b(n)$ where $n - b(n) = \Omega(n)$ that satisfies the third condition. Finally, we choose $a(n) = (b(n)+n)/2$. □

7. CONCLUSION

A new technique for runtime analysis of EAs has been introduced (Theorem 4), allowing upper bounds on the expected runtime of non-elitist EAs with populations to be derived easily. The technique is similar to the existing fitness level technique used to analyse elitist EAs.

Theorem 4 gives conditions on the operators of the algorithm which, when satisfied, can be used to predict when the EA is efficient. The first condition is the same as in the classical fitness level technique. The second condition states that the offspring should with not too small probability be at least as fit as the parent. The third condition prescribes a required level of selective pressure relative to the mutation strength. The final condition sets a lower limit on the population size. The second and fourth conditions can in most cases be verified independently of the fitness function, as shown in Section 4.

The theorem is illustrated on standard example functions. The results, which are summarised in Table 1, are interesting in their own right, showing how the runtime of the EA depends critically on the selective pressure. It has been an open question to quantify the degree of selective pressure necessary for efficient search. It is generally claimed that the selective pressure should not be too strong in order to prevent loss of diversity (i.e. to enable "exploration"), and not be too weak in order to allow the search to be effective (i.e., to enable "exploitation"). Table 1 shows parameter settings that ensure the necessary selective pressure needed to optimise a range of classical example functions efficiently. For the considered selection mechanisms, except fitness proportionate selection, there are sharp, non-linear transitions in parameter space between parameter settings that lead to exponential runtime, and parameter settings that lead to polynomial runtime.

Note that the upper bounds on the runtimes in Table 1 are weaker than the corresponding bounds for the (1+1) EA. This means that the (1+1) EA may perform better on the chosen functions, or that Theorem 4 may be improved.

In general, the paper addresses the lack of techniques available for analysing the runtime of population-based EAs. The new technique reduces the gap between the complexity of algorithms used in practice, and the simplicity of the algorithms amenable to rigorous analysis. Together with the lower bound technique provided in [10], it is a step towards a better understanding of population-dynamics in EAs.

8. REFERENCES

[1] T. Chen, J. He, G. Sun, G. Chen, and X. Yao. A new approach for analyzing average time complexity of population-based evolutionary algorithms on unimodal problems. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(5):1092–1106, Oct. 2009.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. McGraw Hill, New York, NY, 2nd edition, 2001.

[3] S. Droste, T. Jansen, and I. Wegener. On the analysis of the (1+1) Evolutionary Algorithm. *Theoretical Computer Science*, 276:51–81, 2002.

[4] D. E. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of Genetic Algorithms*, pages 69–93. Morgan Kaufmann, 1991.

[5] E. Happ, D. Johannsen, C. Klein, and F. Neumann. Rigorous analyses of fitness-proportional selection for optimizing linear functions. In *Proc. of the 10th annual conference on Genetic and evolutionary computation (GECCO 2008)*, pages 953–960, New York, NY, USA, 2008. ACM.

[6] J. He and X. Yao. Drift analysis and average time complexity of evolutionary algorithms. *Artificial Intelligence*, 127(1):57–85, March 2001.

[7] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[8] T. Kötzing, F. Neumann, D. Sudholt, and M. Wagner. Simple max-min ant systems and the optimization of linear pseudo-boolean functions. To appear in Proc. of Foundations of Genetic Algorithms (FOGA 2011), 2011.

[9] J. Lässig and D. Sudholt. General scheme for analyzing running times of parallel evolutionary algorithms. In *Proc. of the 11th international conference on Parallel problem solving from nature: Part I*, (PPSN 2010), pages 234–243, Berlin, Heidelberg, 2010. Springer-Verlag.

[10] P. K. Lehre. Negative drift in populations. In *Proc. of the 11th international conference on Parallel problem solving from nature: Part I*, (PPSN 2010), pages 244–253, Berlin, Heidelberg, 2010. Springer-Verlag.

[11] P. K. Lehre and X. Yao. On the impact of mutation-selection balance on the runtime of evolutionary algorithms. *To appear in IEEE Trans. on Evolutionary Computation*, 2011.

[12] F. Neumann, P. S. Oliveto, and C. Witt. Theoretical analysis of fitness-proportional selection: landscapes and efficiency. In *Proc. of the 11th Annual conference on Genetic and evolutionary computation (GECCO 2009)*, pages 835–842, New York, NY, USA, 2009. ACM.

[13] P. S. Oliveto, J. He, and X. Yao. Time complexity of evolutionary algorithms for combinatorial optimization: A decade of results. *Intl. Journal of Automation and Computing*, 4(1):100–106, 2007.

[14] D. Sudholt. General lower bounds for the running time of evolutionary algorithms. In *Proc. of the 11th international conference on Parallel problem solving from nature: Part I*, (PPSN 2010), pages 124–133, Berlin, Heidelberg, 2010. Springer-Verlag.

[15] I. Wegener and C. Witt. On the analysis of a simple evolutionary algorithm on quadratic pseudo-boolean functions. *Journal of Discrete Algorithms*, 3(1):61–78, 2005.

[16] C. Witt. Runtime Analysis of the $(\mu + 1)$ EA on Simple Pseudo-Boolean Functions. *Evolutionary Computation*, 14(1):65–86, 2006.

APPENDIX

THEOREM 15 (HOEFFDING'S INEQUALITY [7]). *If $X = \sum_{i=1}^{m} X_i$, where $X_i \in [0, b], i \in [m]$, are indep. random variables, then $\Pr(X - \mathbf{E}[X] \geq t) \leq \exp(-2t^2/b^2m)$ for $t > 0$.*

THEOREM 16 (DRIFT THEOREM [6]). *Let $X_1, X_2, \ldots$ be a stochastic process over S, and $d : S \to \mathbb{R}_0^+$ a distance function on S. Define T to be the first time t such that $d(X_t) = 0$. If there exists a constant $\Delta > 0$ such that,*

1. *$\forall t \geq 0 : \Pr(d(X_t) < B) = 1$, and*
2. *$\forall t \geq 0 : \mathbf{E}[d(X_t) - d(X_{t+1}) \mid T > t] \geq \Delta$,*

then $\mathbf{E}[T] \leq B/\Delta$.

LEMMA 17. *For all $x \in \mathbb{R}$, $1 - e^{-x} \geq x/(x + 1)$.*

PROOF. From $e^x \geq x + 1$, it follows that $1 - 1/e^x \geq (x + 1)/(x + 1) - 1/(x + 1) = x/(x + 1)$. □

LEMMA 18. *If $X \sim \text{Bin}(\lambda, p)$ with $p \geq (i/\lambda)(1 + \delta)$, then $\mathbf{E}\left[e^{-\kappa X}\right] \leq e^{-\kappa i}$ for any $\kappa \in (0, \delta)$.*

PROOF. The value of the moment generating function $M_X(t)$ of the binomially distributed variable X at $t = -\kappa$ is

$$\mathbf{E}\left[e^{-\kappa X}\right] = M_X(-\kappa) = (1 - p(1 - e^{-\kappa}))^{\lambda}. \qquad (8)$$

By Lemma 17 and $1 + \kappa < 1 + \delta$, it follows that

$$p(1 - e^{-\kappa}) \geq \frac{i(1 + \delta)}{\lambda}\left(\frac{\kappa}{1 + \kappa}\right) \geq \frac{\kappa i}{\lambda}.$$

From (8), we now have $\mathbf{E}\left[e^{-\kappa X}\right] \leq (1 - \kappa i/\lambda)^{\lambda} \leq e^{-\kappa i}$. □