TECHNISCHE UNIVERSITÄT DORTMUND

REIHE COMPUTATIONAL INTELLIGENCE

COLLABORATIVE RESEARCH CENTER 531

Design and Management of Complex Technical Processes and Systems by means of Computational Intelligence Methods

A Mix of Markov-Chain and Drift Analysis

Jens Jägersküpper

No. CI-250/08

Technical Report ISSN 1433-3325 June 2008

Secretary of the SFB 531 · Technische Universität Dortmund · Dept. of Computer Science/LS 2 · 44221 Dortmund · Germany

This work is a product of the Collaborative Research Center 531, "Computational Intelligence," at the Technische Universität Dortmund and was printed with financial support of the Deutsche Forschungsgemeinschaft.

A Mix of Markov-Chain and Drift Analysis

Jens Jägersküpper*

Technische Universität Dortmund, Informatik 2, 44221 Dortmund, Germany

Abstract In their seminal article [Theo. Comp. Sci. 276(2002):51–82] Droste, Jansen, and Wegener present the first theoretical analysis of the expected runtime of a basic direct-search heuristic with a global search operator, namely the (1+1) Evolutionary Algorithm ((1+1) EA), for the class of linear functions over the search space $\{0,1\}^n$. In a rather long and involved proof they show that, for any linear function, the expected runtime of the (1+1) EA is $O(n \log n)$, i. e., that there are two constants c and n' such that, for $n \geq n'$, the expected number of iterations until a global optimum is generated is bound above by $c \cdot n \log n$. However, neither c nor n' are specified – they would be pretty large. Here we reconsider this optimization scenario to demonstrate the potential of an analytical method that makes use not only of the drift (w. r. t. a potential function, here the number of bits set correctly), but also of the distribution of the evolving candidate solution over the search space $\{0,1\}^n$: An invariance property of this distribution is proved, which is then used to derive a significantly better lower bound on the drift. Finally, this better estimate of the drift results in an upper bound on the expected number of iterations of $3.8\, n \log_2 n + 7.6 \log_2 n$ for $n \geq 2$.

1 Introduction

We consider the optimization of a pseudo-Boolean function $f\colon \{0,1\}^n \to \mathbb{R}$ which is given by a black box. That is, knowledge about f can solely be gathered by evaluating f at a number of search points. Black-box optimization is also commonly referred to as *direct search.* Randomized neighborhood search is a commonly used heuristic for direct search in $\{0,1\}^n$. It is an iterative method which tries (in each iteration) to pick a better solution from the neighborhood of the current candidate solution by choosing one of the neighbors uniformly at random. Usually, the neighborhood of $x \in \{0,1\}^n$ consists of the Hamming neighbors of x, i. e. of all $y \in \{0,1\}^n$ with a Hamming distance $H(x,y) = 1$ (number of bits that differ). This heuristic is often called randomized local search (RLS). Obviously, RLS cannot escape a local optimum x^* that is not globally optimal and for which all neighbors have a worse function value. Naturally, a different neighborhood could be chosen. As nothing is known about f, however, this is hard choice to make. A different approach is to consider the complete search space as the neighborhood, but to not sample uniformly at random any longer. One of these so-called global methods is the (1+1) EA. It starts with a search point (called *individual*) which is uniformly chosen from $\{0,1\}^n$. Then, in each iteration, a new candidate solution is generated by *mutation,* namely by independently flipping each bit of the current

* supported by the German Research Foundation (DFG) through the collaborative research center „Computational Intelligence" (SFB 531)

candidate solution with a predefined probability p. Usually, $p := 1/n$ is chosen, and this *mutation rate* will also be considered here in the following. Iff the f-value of the mutant is at least as good, then the mutant becomes the next iterate (otherwise the mutant is discarded, so that the search stays where it was), which is called *elitist selection.*

Note that global convergence is trivially proved for the (1+1) EA: Since in each iteration a global optimum is sampled with probability at least p^n, the expected number of steps until a global optimum is generated is bounded above by p^{-n}, namely by n^n for $p = 1/n$. So (global) convergence is not the point. The point is: How long does it actually take? That is: What is the expected number of steps until a global optimum is generated? It turned out that, because of its global search operator, the (1+1) EA is often much harder to analyze than RLS. After first analyses (for simple functions like ONEMAX) using Markov chain theory, cf. [1], a different analytical approach from the field of classical algorithmics enabled a bunch of results for less trivial function scenarios: the potential method. Instead of considering the f-value (of the evolving individual), a potential function is defined (w. r. t. the process), which takes its maximum value if and only if the f-value is best. In many such analyses of the (1+1) EA, the number of bits set correctly is considered as the potential (of a search point). This technique was used by Droste, Jansen, and Wegener [2] to prove, among other results, that the expected number of iterations the (1+1) EA needs to generate the optimum of a linear function is $O(n \log n)$. The expected change in the potential (per iteration) is commonly called *drift.* Drift analysis has been put forward by He and Yao [3], for instance. Unfortunately, the application of their quite general approach to the scenario considered here, namely the (1+1) EA using the standard mutation rate $p = 1/n$ to maximize a linear function, contains a flaw [4], so that this fundamental scenario is not covered. Here we reconsider this scenario and show how to prove a significantly better lower bound on the potential's drift using an invariance property of the distribution of the evolving individual over the search space $\{0,1\}^n$ during the optimization process.

The scenario: We consider the class of linear functions over $\{0,1\}^n$ consisting of all $f : \{0,1\}^n \to \mathbb{R}$ with $f(x) := c + \sum_{i=1}^{n} c_i \cdot x_i$, where $x = x_n \cdots x_1$. We assume that f depends essentially on all n bits, i. e., we assume $c_i \in \mathbb{R} \setminus \{0\}$. The coefficient c_i will also be called *the weight* of the ith bit. Solely for better legibility, we assume $c_n \geq c_{n-1} \geq \cdots \geq c_1 > c = 0$. Obviously, the ordering of the bits is irrelevant. Moreover, because of the uniformly random initialization and the invariance property of the mutation operator of the (1+1) EA (it does not care about whether a bit is 1 or 0), these assumptions can be made without a loss of generality: The (1+1) EA behaves identically on f and $f_{\oplus y}(x) := f(x \oplus y)$ for any fixed $y \in \{0,1\}^n$, in particular for y the complement of the optimum ("$\oplus$" denotes the bitwise XOR operation).

Two linear functions which are frequently considered are ONEMAX, where $c_i := 1$ for each weight, and BINVALUE, where the bit-string is taken as the binary representation, i. e., $c_i := 2^{i-1}$. These are two extremes in the class of linear functions: In ONEMAX all bits have the same weight, whereas in BINVALUE the weight of a single bit (namely the nth) is larger than the total weight of the other $n-1$ bits.

Unless stated differently, we consider the maximization of a linear function with positive weights (non-decreasing from left to right; as described above) by the (1+1) EA (as described above) using the standard mutation rate (bit-flip probability) $p := 1/n$.

2 Invariance property of the individual's distribution over $\{0,1\}^n$

In this section a particular – actually intuitive – property of the distribution of the evolving individual x over the search space $\{0,1\}^n$ will be proved. This will be used in the following Section 3 to prove that *bad mutations,* namely mutations that would result in a loss of 1-bits if they were accepted, are likely to be such that they are discarded by elitist selection. In Section 4 this observation will enable us to prove a significantly better lower bound on the drift (expected change of the number of 1-bits), which is then used to obtain our main result, a better upper bound on the expected optimization time.

Now, consider two bits x_i and x_j in x with $j > i$ (not necessarily adjacent), so that $c_j \geq c_i$ for their weights. First consider the case $x_j x_i = 00$. Assume that a mutation flips x_i, but not x_j, and maybe some more bits. If this mutation is accepted, then the mutation that flips the same bits except for x_j instead of x_i would also be accepted. Note that the latter mutation occurs with the same probability. Now consider the case $x_j x_i = 11$ and assume that a mutation flips x_j, but not x_i, and maybe some more bits. If this mutation is accepted, then the mutation that flips the same bits except for x_i instead of x_j would also be accepted. And again, the latter mutation occurs with exactly the same probability. Thus, since $c_j \geq c_i$, for x_j a change from 0 to 1 is at least as probable as for x_i, whereas a change from 1 to 0 is at most as probable. All in, observing $x_j = 1$ seems at least as probable as observing $x_i = 1$ during the optimization – whatever the number of iterations. This can be formulated more formally:

Theorem 1. *Let $x^{[t]}$ denote the random individual (distributed over $\{0,1\}^n$) after t iterations in our scenario. Then* $\mathrm{Prob}(x_n^{[t]} = 1) \geq \ldots \geq \mathrm{Prob}(x_1^{[t]} = 1)$ *for* $t \geq 0$.

This invariance property of the distribution of the evolving individual over $\{0,1\}^n$ will be proved in the remainder of this section. The superscript in "$x^{[t]}$" will be dropped (unless necessary). Note that „$\mathrm{Prob}(x_j = 1) \geq \mathrm{Prob}(x_i = 1)$“ is equivalent to „$\mathrm{Prob}(x_j x_i = 10) \geq \mathrm{Prob}(x_j x_i = 01)$“ since

$$\mathrm{Prob}(x_j = 1) = \mathrm{Prob}(x_j x_i = 10) + \mathrm{Prob}(x_j x_i = 11)$$
$$\mathrm{Prob}(x_i = 1) = \mathrm{Prob}(x_j x_i = 01) + \mathrm{Prob}(x_j x_i = 11)$$

Thus, to prove our conjecture we merely have to show that for any pair of adjacent bits in $x \in \{0,1\}^n$, i. e. for $i \in \{0,\ldots,n-2\}$

$$\sum_{X\in\{0,1\}^i, Y\in\{0,1\}^{n-2-i}} \mathrm{Prob}(x = X10Y) \geq \sum_{X\in\{0,1\}^i, Y\in\{0,1\}^{n-2-i}} \mathrm{Prob}(x = X01Y)$$
$$\iff \sum_{X\in\{0,1\}^i, Y\in\{0,1\}^{n-2-i}} \big(\mathrm{Prob}(x = X10Y) - \mathrm{Prob}(x = X01Y)\big) \geq 0 \qquad (1)$$

"$XY \in \{0,1\}^j$" abbreviates "$X, Y \in \{0,1\}^* \wedge |X| + |Y| = j$" in the following. Then

$$(\forall XY \in \{0,1\}^{n-2})\ \mathrm{Prob}(x = X10Y) - \mathrm{Prob}(x = X01Y) \geq 0$$

is sufficient for each of the $n-1$ sums in Eqn. (1) to be non-negative. Thus, our intermediate objective is to prove that for the evolving individual x

$$(\forall XY \in \{0,1\}^{n-2})\ \mathrm{Prob}(x^{[t]} = X10Y) \geq \mathrm{Prob}(x^{[t]} = X01Y) \qquad (2)$$

throughout the complete optimization process, i. e. for $t \geq 0$. We will use induction on the number t of iterations to prove Eqn. (2). For the induction step, recall that the optimization of the (1+1) EA is a Markov chain with state space $\{0,1\}^n$. The transition probabilities obviously depend on the function f to be maximized. Let $H(\cdot,\cdot)$ denote the Hamming distance between two bit-strings and let $p(h) = p^h(1-p)^{n-h}$ denote the probability that a mutation flips exactly h bits at particlar positions in the string. Then after iteration $t \geq 1$ the Markov chain is in state $y \in \{0,1\}^n$ with probability

$$\begin{aligned}\mathrm{Prob}(x^{[t]} = y) = \mathrm{Prob}(x^{[t-1]} = y) \cdot \sum_{w \in \{0,1\}^n :\ f(w) < f(y)} p(H(y,w)) \qquad (3)\\ + \sum_{z \in \{0,1\}^n :\ f(z) \leq f(y)} \mathrm{Prob}(x^{[t-1]} = z) \cdot p(H(z,y))\end{aligned}$$

The first sum, weighted by $\mathrm{Prob}(x^{[t-1]} = y)$, equals the probability to generate a worse mutant from y, whereas the second sum equals the probability that the mutant of the current state z in the ith step is y, where z can be any state not better than y. This identity will be used in the induction step of the proof of Eqn. (2). Before we do so, however, we take a closer look at the probability to generate a worse mutant.

Lemma 1. *Let* $\mathrm{Mut} : \{0,1\}^n \to \{0,1\}^n$ *denote the random mapping induced by the mutation operator. Then in our scenario for all* $XYZ \in \{0,1\}^{n-2}$*:*
$\mathrm{Prob}\big(f(\mathrm{Mut}(X1Y0Z)) < f(X1Y0Z)\big) \geq \mathrm{Prob}\big(f(\mathrm{Mut}(X0Y1Z)) < f(X0Y1Z)\big)$.

This is almost obvious since $X1Y0Z$ and $X0Y1Z$ have the same number of 1-bits and $f(X1Y0Y) \geq f(X0Y1Z)$. Nevertheless, a proof can be found in the appendix. This result is now used to estimate the transition probabilities in Eqn. (3) within the proof of the following lemma (which implies Theorem 1, the invariance property).

Lemma 2. *In our scenario, after any number of iterations, i. e. for* $t \geq 0$*, the distribution of the evolving individual* $x^{[t]}$ *over* $\{0,1\}^n$ *is such that for all* $XYZ \in \{0,1\}^{n-2}$
$\mathrm{Prob}(x^{[t]} = X1Y0Z) \geq \mathrm{Prob}(x^{[t]} = X0Y1Z)$.

Proof. The induction basis is trivial: Since $x^{[0]}$ is uniformly distributed over $\{0,1\}^n$, $X1Y0Z$ and $X0Y1Z$ are equiprobable after initialization. For the induction step let $XYZ \in \{0,1\}^{n-2}$ be arbitrary, but fixed. Now consider Eqn. (3) for $y := X0Y1Z$ and for $y := X1Y0Z$, telling us the probabilities of the Markov chain being in state $X1Y0Z$ resp. in state $X0Y1Z$ after t iterations (where $p(h) = p^h(1-p)^{n-h}$ is the probability that a mutation flips exactly h particular bits, where $p \in (0, 1/2)$ is the bit-flip probability of the mutation operator). For the induction step we need to show $\mathrm{Prob}(x^{[t]} = X1Y0Z) \geq \mathrm{Prob}(x^{[t]} = X0Y1Z)$. Lemma 1 actually tells us

$$\sum_{u \in \{0,1\}^n :\ f(u) < f(X1Y0Z)} p\big(H(X1Y0Z,u)\big) \geq \sum_{w \in \{0,1\}^n :\ f(w) < f(X0Y1Z)} p\big(H(X0Y1Z,w)\big),$$

the induction hypothesis $\mathrm{Prob}(x^{[t-1]} = X1Y0Z) \geq \mathrm{Prob}(x^{[t-1]} = X0Y1Z)$, so that

$$\begin{aligned}\mathrm{Prob}(x^{[t-1]} = X1Y0Z) \cdot \sum_{u \in \{0,1\}^n :\ f(X1Y0Z) > f(u)} p\big(H(X1Y0Z,u)\big)\\ \geq \mathrm{Prob}(x^{[t-1]} = X0Y1Z) \cdot \sum_{w \in \{0,1\}^n :\ f(X0Y1Z) > f(w)} p\big(H(X0Y1Z,w)\big).\end{aligned}$$

In other words, the probability of being in state $X1Y0Z$ and staying there (during the tth iteration) is at least as large as it is for $X0Y1Z$. It remains to be shown that getting into state $X1Y0Z$ in the tth iteration is at at least as probable as getting into $X0Y1Z$ in that step. Therefore note that flipping a set of i particular bits is more probable than flipping a set of $i+2$ particular bits when using a bit-flip probability of $p \in (0, 1/2)$ because $0 < p < 1/2 \implies (1-p)^{j-i} > p^{j-i} \iff p^i(1-p)^{n-i} > p^j(1-p)^{n-j}$ for $0 \leq i < j \leq n$. We focus on the rest of the summands, namely we are going to show the following sufficient inequality:

$$\sum_{u \in \{0,1\}^n :\ f(u) \leq f(X1Y0Z)} \mathrm{Prob}(x^{[t-1]} = u) \cdot p\big(H(u, X1Y0Z)\big) \quad =: S_{10}$$

$$\geq \sum_{w \in \{0,1\}^n :\ f(w) \leq f(X0Y1Z)} \mathrm{Prob}(x^{[t-1]} = w) \cdot p\big(H(w, X0Y1Z)\big) \quad =: S_{01}$$

Note that any index w in S_{01} occurs also as u in S_{10} since $f(w) \leq f(X0Y1Z) \leq f(X1Y0Z)$. In the following $A \in \{0,1\}^{|X|}$, $B \in \{0,1\}^{|Y|}$, $C \in \{0,1\}^{|Z|}$ and $h := H(ABC, XYZ)$. We consider different cases for the summation index w in S_{01}:
$\underline{w = A0B0C:}$ Since $ABC \in \{0,1\}^{n-2}$ such that $f(A0B0C) \leq f(X0Y1Z)$, and since $f(X0Y1Z) \leq f(X1Y0Z)$, also $f(A0B0C) \leq f(X1Y0Z)$, so that $u = A0B0C$ is an index in S_{10}, too. Finally, the Hamming distance of $A0B0C$ from $X0Y1Z$ as well as from $X1Y0Z$ equals $h+1$, respectively, so that the summand associated with the index $A0B0C$ in S_{10} equals the summand associated with $A0B0C$ in S_{10}.
$\underline{w = A1B1C:}$ This case is analogous to the case $w = A0B0C$ above.
$\underline{w = A1B0C:}$ This implies that $w = A0B1C$ is also an index in S_{01}. Furthermore, recall that these two indices necessarily occur in S_{10}, too. Thus, in this case

$$\begin{aligned} &\mathrm{Prob}(x^{[t]} = X0Y1Z \mid x^{[t-1]} \in \{A0B1C, A1B0C\}) \\ &\qquad \leq \mathrm{Prob}(x^{[t]} = X1Y0Z \mid x^{[t-1]} \in \{A0B1C, A1B0C\}) \end{aligned}$$

$$\Leftrightarrow \quad \left.\begin{matrix} \mathrm{Prob}(x^{[t-1]} = A0B1C) \cdot p(h) + \\ \mathrm{Prob}(x^{[t-1]} = A1B0C) \cdot p(h+2) \end{matrix}\right\} \leq \left\{\begin{matrix} \mathrm{Prob}(x^{[t-1]} = A0B1C) \cdot p(h+2) + \\ \mathrm{Prob}(x^{[t-1]} = A1B0C) \cdot p(h) \end{matrix}\right.$$

$$\Leftrightarrow \mathrm{Prob}(x^{[t-1]}{=}A0B1C) \cdot (p(h) - p(h{+}2)) \leq \mathrm{Prob}(x^{[t-1]}{=}A1B0C) \cdot (p(h) - p(h{+}2))$$

where the last inequality holds because of the induction hypothesis and $p(h) > p(h+2)$, i. e., $p(h) - p(h+2) > 0$ as seen above. In other words, the two summands associated with the indices $A0B1C$ and $A1B0C$ result in a total value in S_{10} that is at least as large as the value of the two summands corresponding to these two indices in S_{01}.
$\underline{w = A0B1C \text{ and } f(A1B0C) > f(X0Y1Z):}$ In this case, $A0B1C$ is an index in S_{01}, but $A1B0C$ is not. As $f(A0B1C) \leq f(X0Y1Z) \Rightarrow f(A1B0C) \leq f(X1Y0Z)$, however, in S_{10} not only $u = A0B1C$ is an index, but also $u = A1B0C$ is an index. As $H(A0B1C, X0Y1Z) = H(A1B0C, X1Y0Z)$ and, due to the induction hypothesis, $\mathrm{Prob}(x^{[t-1]} = A0B1C) \leq \mathrm{Prob}(x^{[t-1]} = A1B0C)$, the summand in S_{10} associated with $u = A1B0C$ is at least as large as the summand in S_{01} associated with $w = A0B1C$. (Even more, in contrast to S_{01} where $A1B0C$ is not an index, in S_{10} there is an additional summand associated with $u = A0B1C$.)

All in all, in S_{10} there are as many summands as in S_{01} and they sum to an over-all value at least as large as the value of S_{01}. As seen above, this finishes the induction. □

It is easily seen (using the argument given right after Theorem 1) that this result directly implies Theorem 1, the invariance property of the evolving individual's distribution.

3 On the probability of bad mutations

Now, this result on the distribution of the ones in the evolving individual allows us to obtain better bounds on the drift in the potential, namely the number of 1-bits in the individual. For instance, in the case when a mutation flips two bits x_i and x_j, where $j > i$ (so that $c_j \geq c_i > 0$ for their weights), then $\mathrm{Prob}(x_j = 1 \wedge x_i = 0) \geq \mathrm{Prob}(x_j = 0 \wedge x_i = 1)$. This holds for arbitrarily chosen i, j such that $n \geq j > i \geq 1$. Thus, whenever a mutation flips exactly two bits, then the sub-string flipped is „10“ as least as probable as it is „01“. For shorter notation, we introduce the following notions:

Definition 1. *Consider the mutation of a given individual $x \in \{0,1\}^n$.*

***B*-mutation** *Let $B \in \{0,1\}^*$ be the substring in/of x consisting of the bits chosen to be mutated/flipped. Then we call the mutation of x a "B-mutation."*

***z*-zeros-*k*-ones mutation** *A mutation that flips exactly z zeros and exactly k ones in x.*

***z*-zeros mutation** *A mutation that flips exactly z zeros (and possibly some ones) in x.*

surely unacceptable mutation *A mutation that flips more ones than zeros such that each flipping 0-bit can be mapped one-to-one to a flipping 1-bit with a weight at least as large as the weight of the associated zero, respectively.*

potentially acceptable mutation *A mutation that is not surely unacceptable.*

In the example preceding the definition, we noticed that a 1-zero-1-one mutation is at least as probable a 10-mutation as it is a 01-mutation. More general, since the random choice of the bits to be flipped is independent of the individual ('s distribution over $\{0,1\}^n$), we obtain for our scenario as a direct consequence of Lemma 2 the following:

Corollary 1. *Let $J, K, L \in \{0,1\}^*$. In our scenario the mutation observed in an arbitrary but fixed step is at least as probable a $J1K0L$-mutation as a $J0K1L$-mutation.*

To make actual use of this result, consider the relation $R \subset \cup_{k \in \{2,\ldots,n\}} \{0,1\}^k \times \{0,1\}^k$ defined by $(A, B) \in R :\Leftrightarrow (\exists J, K, L \in \{0,1\}^*)\ A = J0K1L \wedge B = J1K0L$. „$(A, B) \in R$“ is written as „$A \leq_R B$“. Furthermore, we let R^* denote the transitive hull of the relation R. Then the above corollary extends to the following:

Corollary 2. *Let $A, B \in \{0,1\}^k$, $2 \leq k \leq n$, such that $A \leq_{R^*} B$. In our scenario, in an arbitrary but fixed step, a B-mutation occurs at least as probable as an A-mutation.*

Our intermediate objective is to show that, whenever a mutation flips more ones than zeros – which would result in a loss of ones if the mutation was accepted – then this bad mutation is rather likely to be an unacceptable one, so that the loss of ones is *not* accepted. By this, the drift w. r. t. the potential (#ones in the individual) is supposed to become larger, hopefully $\Omega(1/\#\text{ zeros})$. The following result will be utilized therefor.

Lemma 3. *In our scenario for $k \geq 2$: Whenever a 1-zero-k-ones mutation occurs in a step, then this mutation is accepted at most with probability $1/(k+1)$.*

Proof. As the bits' weights decrease from left to right, R^* induces a total order on the 1-zero-k-ones mutations because $01^k \leq_R 101^{k-1} \leq_R 1101^{k-2} \leq_R \ldots \leq_R 1^k0$. For $k \geq 2$, a 1-zero-k-ones mutation is potentially acceptable only if it is an 01^k-mutation. The other k of the $\binom{k+1}{1} = k+1$ different mutation types are surely unacceptable. Using the preceding corollary, we know that each of these surely unacceptable types occurs at least as probable as a 01^k-mutation, so that the latter occurs at most with a probability of $1/(k+1)$ (given that a 1-zero-k-ones mutation occurs). □

Analogous results for mutations that flip two (or more) 0-bits can be obtained. For our scenario, however, 1-zero mutations are the crucial ones, so that we focus on these.

4 Better estimate of the drift – better bound on the runtime

Let Γ denote the power-set of $\{1, \ldots, n\}$ and $p(b) := p^b(1-p)^{n-b}$ the probability that a mutation flips exactly b bits. Let $\mathrm{Mut}(x, I)$ denote the mutant obtained by flipping the bits in $x \in \{0,1\}^n$ that are determined by the index set $I \in \Gamma$. Then the drift equals

$$\sum_{I \in \Gamma} p(\#I) \cdot \big(\#\{i \in I \mid x_i = 0\} - \#\{i \in I \mid x_i = 1\}\big) \cdot [f(\mathrm{Mut}(x,I)) \geq f(x)]\,, \quad (4)$$

where $[\cdot]$ is an indicator variable that resolves to 1 if the predicate is true, and to 0 otherwise. In the following, the summands will be grouped according to the number $z \in \{0, \ldots, n\}$ of zeros that are flipped. If a z-zeros-k-ones mutation is accepted, the number of ones changes by $z - k$. The probability that exactly z zeros and k ones are flipped in $x \in \{0,1\}^n$ equals $\boldsymbol{P_{x,p}(z,k)} := \binom{|x|_0}{z} \cdot \binom{|x|_1}{k} \cdot p^{z+k} \cdot (1-p)^{n-(z+k)}$. For $k > z \geq 1$, let $\boldsymbol{A_{x,p}(z,k)}$ denote an upper bound on the probability that a z-zeros-k-ones mutation of $x \in \{0,1\}^n$ is accepted. Let $z \geq 1$ be fixed. For $k > z$, the summands in Eqn. (4) for which I corresponds to a z-zeros-k-ones mutation sum up to a negative value not smaller than $P_{x,p}(z,k) \cdot A_{x,p}(z,k) \cdot (z-k)$. A z-zero-0-ones mutation is always accepted and increases the number of ones by z (≥ 1). All summands for which I corresponds to a z-zeros-0-ones mutation sum up to $P_{x,p}(z,0) \cdot z$. Thus, for the fixed number $z \geq 1$ of flipping zeros, the contribution of all possible z-zero mutations to the drift (the expected change in the number of 1-bits) is at least

$$\Delta_{x,p}(z) := P_{x,p}(z,0) \cdot z + \textstyle\sum_{z<k\leq|x|_1} P_{x,p}(z,k) \cdot A_{x,p}(z,k) \cdot (z-k)\,. \quad (5)$$

Up to now, $z \geq 1$ was assumed. For $z = 0$, note that in our scenario a mutation that does not flip a 0-bit cannot change the individual, so that the total drift is bounded below by $\boldsymbol{\Delta_{x,p}} := \sum_{z=1}^{|x|_0} \Delta_{x,p}(z)$. For $z \geq 1$ the formula for $\Delta_{x,p}(z)$ can be transformed into

$$\begin{aligned}
& z \cdot \Big(P(z,0) + \sum_{z<k\leq|x|_1} P(z,k) \cdot A(z,k) \cdot \frac{z-k}{z}\Big) \\
&= z \cdot \binom{|x|_0}{z} \cdot \Big(p^z \cdot (1-p)^{n-z} + \sum_{z<k\leq|x|_1} \binom{|x|_1}{k} \cdot p^{z+k} \cdot (1-p)^{n-(z+k)} \cdot A(z,k) \cdot \frac{z-k}{z}\Big) \\
&= z \cdot \binom{|x|_0}{z} \cdot p^z(1-p)^{n-z} \cdot \Big(1 + \sum_{z<k\leq|x|_1} \tbinom{|x|_1}{k} \cdot \Big(\frac{p}{1-p}\Big)^k \cdot A(z,k) \cdot \frac{z-k}{z}\Big)\,. \quad (6)
\end{aligned}$$

The sign of $\Delta_{x,p}(z)$ is determined by the sign of the rightmost factor (in big parentheses). We now show that $\boldsymbol{\Delta_x(z)} := \Delta_{x,1/n}(z)$ is non-negative for all z when $p = 1/n$.

Lemma 4. *In our scenario, where the mutation rate $p = 1/n$ is used, for any fixed individual $x \in \{0,1\}^n$: $\Delta_x(z) \geq 0$ for $z \in \{1, \ldots, |x|_0\}$.*

Proof. Obviously, $z \cdot \binom{|x|_0}{z} \cdot p^z(1-p)^{n-z} \geq 0$, so that we have to show that the rightmost factor $(1 + \sum \ldots)$ in Eqn. (6) is non-negative. $A(z,k) := 1$ is a trivial upper bound on the probability that a (bad) mutation is accepted, so that we concentrate on

$$\sum_{z<k\leq|x|_1} \binom{|x|_1}{k} \cdot \left(\frac{p}{1-p}\right)^k \cdot \frac{-(z-k)}{z} \leq 1. \tag{7}$$

We take a closer look at the summands: $\left(\frac{p}{1-p}\right)^k = \left(\frac{1}{n\cdot(1-1/n)}\right)^k = (n-1)^{-k}$ and $\binom{|x|_1}{k} \leq \frac{(n-1)^k}{k!}$ since $|x|_1 =\leq n-1$. Thus, the sum in Eqn. (7) is bounded above by

$$\sum_{z<k\leq|x|_1} \frac{(n-1)^k}{k!} \cdot (n-1)^{-k} \cdot \frac{k-z}{z} = \sum_{z<k\leq|x|_1} \frac{k-z}{k!\cdot z} \leq \sum_{k\geq 2} \frac{k-1}{k!} \tag{8}$$

since $z \geq 1$ is assumed. The rightmost sum equals $\sum_{k\geq 2}\left(\frac{1}{(k-1)!} - \frac{1}{k!}\right) = \frac{1}{(2-1)!} = 1$, which proves the inequality Eqn. (7) and with it the claimed inequality $\Delta_x(z) \geq 0$. □

So, now we know that $\Delta_x(z)$ is non-negative for any number z of zeros that may be flipped by a mutation – whatever the mutated individual x. Consequently, we know that throughout the optimization process the drift is non-negative in each iteration. For a good upper bound on the runtime, however, we need a positive lower bound on the drift. In fact, the larger the lower bound on the drift, the better. Since for $p = 1/n$ the expected number of bits that flip equals one, we take a second look at $\Delta_x(1)$ to derive a better estimation for the contribution of 1-zero mutations to the drift. In Eqn. (6) the sum over k has been estimated for $z = 1$ using the trivial estimate $A_{x,p}(z,k) = 1$. Now, making use of our knowledge about the distribution of the evolving individual x over $\{0,1\}^n$, namely by Lemma 3, we know that for $k \geq 2$ a 1-zero-k-ones mutation is accepted at most with probability $1/(k+1)$, so that for our scenario $A(1,k) := 1/(k+1)$ can be used instead of the distribution-independent/trivial upper bound $A_{x,p}(1,k) = 1$.

Lemma 5. *For $p := 1/n$, $A(1,k) := 1/(k+1)$, $z := 1$ the following inequality holds:* $\sum_{z<k\leq n} \binom{|x|_1}{k} \cdot \left(\frac{p}{1-p}\right)^k \cdot A(z,k) \cdot \frac{k-z}{z} < 0.282$.

Proof. As in proof of the previous lemma, the sum to be bounded from above is at most $\sum_{z<k\leq n} \frac{1}{k!} \cdot A(z,k) \cdot \frac{k-z}{z}$. For the settings of the lemma we obtain

$$\begin{aligned}\sum_{z<k\leq n} \frac{1}{k!} \cdot A(z,k) \cdot \frac{k-z}{z} &\leq \sum_{k\geq 2} \frac{1}{k!} \cdot \frac{1}{k+1} \cdot (k-1) = \sum_{k\geq 2} \frac{k-1}{(k+1)!} \\ &= \sum_{k\geq 3} \frac{k-2}{k!} = \sum_{k\geq 2} \frac{1}{k!} - \sum_{k\geq 3} \frac{2}{k!} = (\mathrm{e}-2) - 2(\mathrm{e}-2.5)\end{aligned}$$

using $\sum_{k\geq 1} 1/k! = \mathrm{e} - 1$. Finally, $(\mathrm{e}-2) - 2(\mathrm{e}-2.5) = 3 - \mathrm{e} < 0.282$. □

Plugging the estimate of the preceding Lemma 5 into Eqn. (6) for $z := 1$, we obtain

$$\Delta(1) \geq |x|_0 \cdot \frac{1}{n}\left(1-\frac{1}{n}\right)^{n-1} \cdot (1-0.282) \geq \frac{|x|_0}{n} \cdot \frac{0.718}{\mathrm{e}} > \frac{|x|_0}{n} \cdot 0.264$$

(Recall that $\Delta(z) \geq 0$ for all z.) This better estimation of the contribution of 1-zero mutations to the drift utilizing the individual's distribution over the search space $\{0,1\}^n$ is crucial: Now the lower bound on the drift is $\Delta \geq \Delta(1) \geq \frac{|x|_0}{n} \cdot 0.264 = \Omega(\#\text{zeros}/n)$. When we consider the number of zeros as the approximation error (the Hamming distance from the optimum), then the result on the drift reads: As long as there are zeros in x, in each step we expect the approximation error to decrease by a factor smaller (i. e. better) than $1 - 0.264/n$. Since $(1-0.264/n)^{(n/0.264)\cdot \ln 2} \lesssim 1/2$, the number of steps until we expect the progress to be such that the approximation error is halved is less than $(n/0.264) \cdot \ln 2 < 2.63n$. The actual question is, however: What is the expected number of steps to actually halve the approximation error? The following lemma helps us to turn our lower bound on the drift into an upper bound on the expected runtime:

Lemma 6. *Let $X_1, X_2, \ldots$ denote random variables with bounded support and S the random variable defined by $S := \min\{\, t \mid X_1 + \cdots + X_t \geq g\}$ for a given $g > 0$. Given that S is a stopping time (i. e., the event $\{S = k\}$ depends solely on $X_1, \ldots, X_k$), if $\mathsf{E}[S] < \infty$ and $\mathsf{E}[X_i \mid S \geq i\,] \geq \ell > 0$ for all i, then $\mathsf{E}[S] \leq \mathsf{E}[X_1 + \cdots + X_S]/\ell$.*

Proof. Note that the X_i need not be independent and that, since the X_i are bounded, the precondition $\mathsf{E}[S] < \infty$ implies $\mathsf{E}[X_1 + \cdots + X_S] < \infty$. We use

$$\mathsf{E}[X_1 + \cdots + X_S] = \sum_{i=1}^{\infty} \mathrm{Prob}\{S \geq i\,\} \cdot \mathsf{E}[X_i | S \geq i\,] \geq \sum_{i=1}^{\infty} \mathrm{Prob}\{S \geq i\,\} \cdot \ell = \mathsf{E}[S] \cdot \ell$$

where the first equation is the major part of the proof of Wald's equation (a proof can be found in [5, Apx. B] for instance). □

We concentrate on the expected number of steps to halve the approximation error, and thus, in the application of the preceding lemma we let X_i denote the increase in the number of ones in the ith iteration and choose $g := z/2$ and $\ell := (z/2) \cdot 0.264/n$, where we use that $0 \leq X_i \leq n$ in our scenario, and that the condition $\{S \geq i\}$ merely means that the approximation error has not been halved within the first $i-1$ iterations. Finally, we use $\mathsf{E}[X_1 + \cdots + X_S] \leq z/2 + z/n$, where "$+z/n$" is a rough general upper bound on the expected increase in the number of ones (in a step), namely the expected number of flipping zeros, which is $z \cdot p = z/n$ since $p = 1/n$. Thus, the application of the previous lemma yields the following upper bound on the expectation of the number S of steps to halve the approximation error, i. e. the number of 0-bits:

$$\mathsf{E}[S] \leq \frac{\mathsf{E}[X_1 + \cdots + X_S]}{\ell} \leq \frac{z/2 + z/n}{(z/2) \cdot 0.264/n} \leq 3.79n + 7.58.$$

With this bound on the zeros' expected half-life we can finally derive our main result.

Theorem 2. *Let the (1+1) EA using the mutation rate (bit-flip probability)* $1/n$ *maximize a linear function* $f\colon \{0,1\}^n \to \mathbb{R}$*,* $n \geq 2$*. Then the expected number of steps until the evolving individual has maximum* f*-value is smaller than* $3.8\, n \log_2 n + 7.6 \log_2 n$*.*

Proof. Without loss of generality we may assume that all coefficients are positive, so that the all-ones string has maximum function value. As the expected number of 0-bits in the initial individual equals $n/2$, after $\lfloor \log_2(n/2) \rfloor + 1 \leq \log_2 n$ halvings (in expectation w. r. t. the initialization) there is less than one 0-bit left, i. e., the optimal all-ones bit-string has been generated. Since the expected number of iterations to halve the number of zeros is smaller than $3.8n + 7.6$ (independently of the initialization), we obtain an upper bound of $(3.8n + 7.6) \cdot \log_2 n$ on the expected number of iterations. □

5 Conclusions

We have exemplarily shown for the fundamental scenario "standard (1+1) EA on linear functions" how knowledge about the distribution of the evolving individual over the search space can significantly improve (upper) bounds on the expected runtime until an optimum is generated. The invariance property of the individual's distribution proved and then utilized here is actually quite intuitive. Interestingly, its proof is quite straightforward and not that sophisticated. Nonetheless, it enables a remarkable improvement of the estimation of the drift (and, thus, of the runtime). For other scenarios, a suitable invariance property may be harder to find – experiments may actually give useful hints – and probably even harder to prove. But, as we have demonstrated here, these efforts may indeed pay off. Actually, in some situations such knowledge is necessary to obtain an asymptotically tight bound. Then the use of an invariance property of the evolving individual's distribution over the search space can actually be an elegant tool.

Naturally, for more advanced (evolutionary) algorithms that use a population, the distribution of the population over the search space may be considered. In [6] a similar approach is followed for the analysis of a $(\mu{+}1)$ evolution strategy, showing that this technique can indeed make sense for the analysis of population-based algorithms.

References

1. Rudolph, G.: Finite Markov chain results in evolutionary computation: A tour d'horizon. Fundamenta Informaticae **35** (1998) 67–89
2. Droste, S., Jansen, T., Wegener, I.: On the analysis of the (1+1) evolutionary algorithm. Theoretical Computer Science **276** (2002) 51–82
3. He, J., Yao, X.: Drift analysis and average time complexity of evolutionary algorithms. Artificial Intelligence **127** (2001) 57–85
4. He, J., Yao, X.: Erratum to: Drift analysis and average time complexity of evolutionary algorithms [3]. Artificial Intelligence **140** (2002) 245–248
5. Jägersküpper, J.: Algorithmic analysis of a basic evolutionary algorithm for continuous optimization. Theoretical Computer Science **379** (2007) 329–347
6. Jägersküpper, J., Witt, C.: Rigorous runtime analysis of a (μ+1) ES for the Sphere function. In: Proc. 2005 Genetic and Evolutionary Computation Conference (GECCO), ACM Press (2005) 849–856

A Proof of Lemma 1

We have to show that

$$S_{10} := \sum_{\substack{u\in\{0,1\}^n:\\ f(u)<F(A1B0C)}} p(H(X1Y0Z,u)) \;\geq \sum_{\substack{w\in\{0,1\}^n:\\ f(w)<F(A0B1C)}} p(H(X0Y1Z,w)) \;=: S_{01}$$

where $H(\cdot,\cdot)$ denotes the hamming distance and $p(j)$ the probability that in a mutation j particular bits are flipped. Note that any index w in S_{01} occurs also as u in S_{10} because $f(w) \leq f(X0Y1Z) \leq F(X1Y0Z)$. In the following $A \in \{0,1\}^{|X|}$, $B \in \{0,1\}^{|Y|}$, $C \in \{0,1\}^{|Z|}$ and $h := H(ABC, XYZ)$.

- $w = A0B0C$: Since $ABC \in \{0,1\}^{n-2}$ such that $f(A0B0C) \leq f(X0Y1Z)$ ($\leq f(X1Y0Z)$), $u = A0B0C$ is an index in S_{10}, too. The Hamming distance of $A0B0C$ from $X0Y1Z$ as well as from $X1Y0Z$ equals $h+1$, respectively, so that the summands associated with the index $A0B0C$ in S_{01} resp. S_{10} have the same value.
- $w = A1B1C$: This case is analogous to the case $w = A0B0C$ above.
- $w = A1B0C$: Since $f(A1B0C) \geq f(A0B1C)$, in this case $w = A0B1C$ is an index in S_{01}, too. Thus, both indices occur also in S_{10}, and they contribute to each of the sums $p(h) + p(h{+}2)$, respectively.
- $w = A0B1C$ and $f(A1B0C) > f(X0Y1Z)$: In this case, $A0B1C$ is an index in S_{01}, but $A1B0C$ is not. Since $f(A0B1C) \leq f(X0Y1Z)$ implies $f(A1B0C) \leq f(X1Y0Z)$, however, in S_{10} not only $u = A0B1C$ is an index (contributing $p(h{+}2)$ to the sum), but also $u = A1B0C$ is an index, contributing $p(h)$ to S_{10}, which equals the contribution of $w = A0B1C$ in S_{01}.

Hence, each w in S_{01} can be mapped one-to-one to an u in S_{10} such that the contribution of u to the value of the sum S_{10} equals the contribution of the corresponding w to the value of S_{01}. □