Analysis of the (1+1) EA on Subclasses of Linear Functions under Uniform and Linear Constraints

Tobias Friedrich
Hasso Plattner Institute
Potsdam, Germany

Timo Kötzing
Hasso Plattner Institute
Potsdam, Germany

Gregor Lagodzinski
Hasso Plattner Institute
Potsdam, Germany

Frank Neumann
School of Computer Science
The University of Adelaide
Adelaide, Australia

Martin Schirneck
Hasso Plattner Institute
Potsdam, Germany

ABSTRACT

Linear functions have gained a lot of attention in the area of run time analysis of evolutionary computation methods and the corresponding analyses have provided many effective tools for analyzing more complex problems. In this paper, we consider the behavior of the classical (1+1) Evolutionary Algorithm for linear functions under linear constraint. We show tight bounds in the case where both the objective and the constraint function is given by the OneMax function and present upper bounds as well as lower bounds for the general case. We also consider the LeadingOnes fitness function.

Categories and Subject Descriptors

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

Keywords

Run time analysis, evolutionary algorithm, knapsack, constraints.

1. INTRODUCTION

Evolutionary algorithms have been used in a wide range of application domains such as water distribution network [27, 30], renewable energy [21], supply chain management [20], and software engineering [9, 18]. Their easy application and adaptation to a wide range of engineering problems qualify them for research even without a deep algorithmic background.

Although evolutionary computation is very popular in a large variety of application domains, the theoretical understanding lacks behind its practical success. Over the last 20 years a lot of progress in understanding evolutionary computing techniques has been achieved by studying the run time behavior of evolutionary algorithms which are simpler than the ones used in practice, but still capture the main aspects of the algorithms [1, 12, 23].

At the heart of these investigations have been studies of the classical (1+1) EA for the class of linear functions [6]. Initial investigations considered OneMax [22] as the simplest non-trivial pseudo-Boolean function. Later investigations have been generalized to the whole class of linear functions for which it has been shown in [6] that the (1+1) EA optimizes them in expected time $\Theta(n \log n)$. Further studies investigated the (1+1) EA and linear functions, giving simpler proofs and a more precise analysis by giving the constants hidden in the Θ-notation [4, 29]. Furthermore, linear functions, especially OneMax, have been investigated in dynamic [16] and stochastic settings [5, 8] as well as for other evolutionary computing techniques such as particle swarm optimization [28], ant colony optimization [17] and estimation of distribution algorithms [3].

Maximizing a linear function under a linear constraint is equivalent to the well-known NP-hard knapsack problem. Beyond the worst case, the knapsack problem has been well-studied from an average case and smooth complexity perspective and it has been found that this problem can be solved in (expected) polynomial time for a wide range of these settings [2, 26].

It has been shown in [31] that the expected optimization time of the (1+1) EA on a specific deceptive knapsack instance is exponential. We investigate several subclasses of linear functions under linear constraints where the objective function or the given constraint are of type OneMax. We call a constraint given by OneMax a uniform constraint. The goal of our investigations is to gain an understanding on the working principles of the (1+1) EA for these subclasses. The reader should note that the subclasses under investigation can be solved to optimality in polynomial time by deterministic (greedy) algorithms.

Our findings are summarized in Table 1. We start our investigations by considering OneMax together with a uniform constraint of B (that is, only bit strings with at most B 1-bits are feasible) and show that the (1+1) EA is able to find an optimal solution efficiently (in time $O(n \log n)$, but depending on B potentially faster). Note that OneMax

FOGA '17, January 12 - 15, 2017, Copenhagen, Denmark

ISBN 978-1-4503-4651-1/17/01...$15.00

DOI: http://dx.doi.org/10.1145/3040718.3040728

Constraint	Problem	Expected Optimization Time		
uniform	finding a feasible solution	$O(n\log(n/B))$		Lemma 3
	OneMax	$\Theta(\sqrt{n})$,	if $\lvert B-n/2\rvert < \sqrt{n}$	Theorem 4
		$\Theta(\lvert B-n/2\rvert)$,	if $\sqrt{n} \le \lvert B-n/2\rvert < n/4$	
		$\Theta(n\log(n/B_{\min}))$,	if $n/4 \le \lvert B-n/2\rvert$	
	$(1+\varepsilon)\sum_{i=1}^{B} x_i + \sum_{i=B+1}^{n} x_i$	$O(n^2)$		Theorem 7
	linear functions	$\Omega(n^2)$		Theorem 5
		$O(n^2\log(B\,w_{\max}))$		Theorem 6
	LeadingOnes	$O(n^2\log B)$		Theorem 8
linear	OneMax	exponential		Theorem 9

Table 1: Overview of Results. The expected optimization times of the (1+1) EA on linear functions and LeadingOnes on bit strings of length n under uniform or linear constraint B. In the extreme case of $B = 0$, $O(n\log(n/B))$ is to be read as $O(n\log n)$. $B_{\min} = \min\{B, n-B\}$, $0 < \varepsilon < 1/n$ is a positive real number, $w_{\max} \ge 1$ is the largest weight of the linear function. The table shows that the optimization time on OneMax under uniform constraint is never larger than in the unconstrained case and that there are ranges of B in which it is significantly smaller. On the contrary, there is a general linear function and a uniform constraint B such that the optimization time is in $\Omega(n^2)$. There is a linear constraint such that the (1+1) EA needs exponential time even on OneMax.

with uniform constraints has many global optima: any bit string with B 1-bits is optimal. We modify OneMax by increasing the weight of the first B bit positions to $1+\varepsilon$, for a very small value of ε. This ensures that there is only one global optimum. We show that this function requires $O(n^2)$ fitness evaluations to optimize: after reaching the bound of B bits, lighter bits have to be exchanged for more valuable bits; while still k valuable bits are missing, an improving exchange of bits has a probability of $\Theta(k^2/n^2)$. Thus, the Variable Drift Theorem gives us the overall run time of $O(n^2)$.

Investigating more general functions with a uniform constraint, we show that a general upper bound of $O(n^3)$ holds for all linear objective functions. Furthermore, we show that there is a linear function for which the (1+1) EA takes $\Omega(n^2)$ fitness evaluations. We conjecture a general upper bound of $O(n^2)$ for all linear functions, but for now we content ourselves with showing this bound for the $(1+\varepsilon)$-test function mentioned above.

Finally, we show that LeadingOnes can be optimized in time $O(n^2\log B)$ and that OneMax with a specific linear constraint implies an exponential run time for the (1+1) EA.

We proceed in Section 2 by introducing the algorithm and the class of constrained optimization problems that is subject to our investigations. We consider uniform constraints in Section 3 and linear constraints in Section 4. Finally, we conclude in Section 5.

2. PRELIMINARIES

We consider as search space the collection $\{0,1\}^n$ of bit strings $x = x_1x_2\ldots x_n$ of fixed length n and examine the class of linear functions

$$f(x) = \sum_{i=1}^{n} w_i x_i.$$

We assume all weights w_i to be positive real numbers that are w.l.o.g at least 1 and denote by $w_{\max} = \max_i w_i$ the maximal weight.

We investigate the optimization of f under a linear constraint given by

$$b(x) = \sum_{i=1}^{n} b_i x_i \le B$$

where the weights b_i are positive reals and B is a positive upper bound. We call a function f to be under *uniform constraint* if all b_i are equal to 1; otherwise, we say that it is under *linear constraint*. In order to optimize f under the constraint $b(x) \le B$ we employ the (1+1) Evolutionary Algorithm ((1+1) EA) as given in Algorithm 1.

Algorithm 1: (1+1) EA

```
1 Choose x ∈ {0,1}^n uniformly at random;
2 while stopping criterion not met do
3     y ← flip each bit of x ind. with prob. 1/n;
4     if z(y) ≥ z(x) then x ← y;
```

Here, x denotes the best search point found so far. We also use the symbol x' for the (possibly mutated) offspring *after* selection; $x^{(0)}$ denotes the initial bit string drawn in Step 1.

During the optimization we use a penalty approach for dealing with infeasible solutions by

$$z(x) = f(x) - (n\,w_{\max} + 1) \cdot \max\{0, b(x) - B\}.$$

In this manner we ensure that infeasible solutions have negative fitness value, which guides the search towards the feasible region of the search space. In particular, Algorithm 1 will *never* adopt an infeasible solution in Step 4 after sampling the first feasible bit string.

We study the number of iterations the (1+1) EA needs until it samples an optimal solution for the first time. This is called the *optimization time* of the algorithm; we usually denote this random variable as T. The expected value of this variable $E[T]$ is called the *expected optimization time.*

In order to study the expected optimization time of the (1+1) EA we apply drift analysis as introduced by He and Yao [11]. An auxiliary function called the *potential* maps a bit string to the real axis allowing us to evaluate the expected progress between consecutive rounds. Usually the potential is chosen such that its minimization corresponds to maximizing the fitness. In particular, the potential reaches its minimal value just in case the bit string is optimal. Depending on the type of expected potential decrease we apply one of several drift theorems [4, 11, 13, 15, 25, 29] with the following being the most general.

Theorem 1 (Variable Drift Theorem [13])**.** *Let* $(X^{(t)})_{t\geq 0}$ *be a sequence of random variables over a finite state space* $\{0\} \subsetneq S \subsetneq \mathbb{R}_0^+$. *Define* $s_{\min} = \min\{x \in S \mid x > 0\}$ *and* $s_{\max} = \max\{x \in S\}$. *Furthermore, let* T *be the random variable denoting the first point in time* $t \in \mathbb{N}$ *for which* $X^{(t)} = 0$. *Suppose that there exist a monotonically increasing function* $h\colon \mathbb{R}^+ \to \mathbb{R}^+$ *such that* $1/h$ *is integrable on* $[s_{\min}, s_{\max}]$ *and, for all* $t < T$ *and* $0 \neq s \in S$,

$$E\Big[X^{(t)} - X^{(t+1)} \;\Big|\; X^{(t)} = s\Big] \geq h(s).$$

Then, for all $0 \neq s_0 \in S$,

$$E\Big[T \;\Big|\; X^{(0)} = s_0\Big] \leq \frac{s_{\min}}{h(s_{\min})} + \int_{s_{\min}}^{s_0} \frac{1}{h(s)}\,\mathrm{d}s.$$

As the potential usually is based on properties of the current best solution x, we define some notation regarding bit strings. Let $1 \leq i \leq j \leq n$ be two indices. We let $\overline{x_i}$ denote the negation of the bit x_i, $x_{[i,j]} = x_i x_{i+1} \ldots x_j$ is the substring of all bits from position i to j (including). The number of 1-bits in x is denoted $|x|_1 = \sum_{i=1}^{n} x_i$; conversely, $|x|_0 = n - |x|_1$ is the number of 0-bits. We sometimes also use the term *Hamming weight* for the number of 1s.

During the analysis in the sections below we frequently bound an expected value by some conditional expectation. This technique is justified by the following observation from the law of total expectation.

Lemma 2. *Suppose* X *is a discrete random variable taking values in* $\mathbb{R}_0^+$ *and* $\mathcal{E}$ *an arbitrary event with* $0 < P[\mathcal{E}] < 1$. *Then,* $E[X] \geq E[X \mid \mathcal{E}]\,P[\mathcal{E}]$. *If additionally* $X > 0$ *implies* $\mathcal{E}$*, equality holds.*

Proof. The conditional expectation $E[X \mid \neg\mathcal{E}]$ exists and cannot be negative due to X being non-negative. Hence,

$$\begin{aligned} E[X] &= E[X \mid \mathcal{E}]\,P[\mathcal{E}] + E[X \mid \neg\mathcal{E}]\,P[\neg\mathcal{E}] \\ &\geq E[X \mid \mathcal{E}]\,P[\mathcal{E}]. \end{aligned}$$

If the second condition is met, then $E[X \mid \neg\mathcal{E}] = 0$. □

3. UNIFORM CONSTRAINT

We start with investigating uniform constraints, i.e., $b_i = 1$ for all $1 \leq i \leq n$. This only restricts the total number of 1-bits in a feasible solution. Hence, we assume the weight bound B to be an integer between 0 and n.

In the following lemma we derive a general bound on the time the (1+1) EA on any pseudo-Boolean function under uniform constraint needs to sample a feasible solution. This will ease the later analysis. We reduce the problem at hand to the well-known case of the (1+1) EA on the OneMax function defined as

$$\textsc{OneMax}(x) = \sum_{i=1}^{n} x_i.$$

We would like to point out that throughout this paper run time estimates of the form $\mathrm{O}(n\log(n/B))$ should be read as $\mathrm{O}(n\log n)$ in the extreme case of $B = 0$.

Lemma 3. *Consider the (1+1) EA optimizing an arbitrary non-negative pseudo-Boolean function under uniform constraint* B. *Then, the expected number of iterations until the algorithm samples a feasible solution for the first time is in* $\mathrm{O}(n\log(n/B))$.

Proof. In the infeasible range the (1+1) EA strictly prefers bit strings with fewer 1-bits due to the large penalty term of $(n\,w_{\max}+1)$. Hence, the optimization process equals that on an unconstrained OneMax function considered as a minimization problem, a mutation is accepted if and only if it does not increase the total number of 1-bits $|x|_1$. A standard argument gives an expected drift of at least

$$E[|x|_1 - |x'|_1 \mid |x|_1 > B] \geq \frac{|x|_1}{n}\left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{|x|_1}{en}.$$

by flipping any of the $|x|_1$ 1-bits and nothing else. The Multiplicative Drift Theorem [4] now yields an expected waiting time of

$$E[T] \leq en\left(\ln\Big(\frac{n}{B}\Big) + 1\right)$$

until $|x|_1$ is reduced from at most n below the cardinality constraint B. □

As we will see in the next section the bound established above is not always tight. The importance of this result lies elsewhere. By employing Lemma 3, we will often be able to assume the optimization starts with a feasible solution without affecting the asymptotic run time.

3.1 OneMax

In the infeasible region of the search space any pseudo-Boolean function behaves like a OneMax problem. To complement this, we now examine the optimization process of the (1+1) EA on OneMax as the objective function. The run time turns out to be heavily dependent on the size of the cardinality bound B relative to the length n of the bit string. The following theorem shows that the time needed is never worse than in the unconstrained case. Furthermore, OneMax can be maximized even in sub-linear time if B is close to $n/2$.

The analysis of the optimization benefits extensively from symmetries inherent to the underlying random process. For OneMax The fitness function is invariant under permutations and the mutation operator of the (1+1) EA is indifferent towards the position and the value of the bits. Additionally, the number of 1-bits and the number of 0-bits in the initial solution are identically distributed, where this distribution is symmetric around its mean value $n/2$.

Theorem 4. *Let $B_{\min}$ denote $\min\{B, n-B\}$. The expected optimization time of the (1+1) EA on* OneMax *under uniform constraint B is in*

$$\begin{aligned} &\Theta(\sqrt{n}), && \text{if } \left|B-\frac{n}{2}\right| < \sqrt{n};\\ &\Theta\Big(\left|B-\frac{n}{2}\right|\Big), && \text{if } \sqrt{n} \le \left|B-\frac{n}{2}\right| < \frac{n}{4};\\ &\Theta\Big(n\log\Big(\frac{n}{B_{\min}}\Big)\Big), && \text{otherwise.}\end{aligned}$$

Proof. In this proof we identify the three main quantities that affect the expected run time of the (1+1) EA on constrained OneMax: the expected drift, the distance between the initial number of 1-bits and the cardinality bound B, and how far B is away from the central value $n/2$. Intuitively, the distance of the initial solution to B marks the ground we have to cover until we reach an optimal solution and the drift is the speed we travel with. The difference $|B - n/2|$ partitions the range of all possible values of B into regions corresponding to different asymptotic run times.

The first part of this proof is presented as a series of claims giving bounds on these quantities. While the drift can be inferred from standard arguments (Claim 1 below), the analysis of the influence of the two distance measures is more involved. The initial bit string has Hamming weight $n/2$ in expectation. Counter-intuitively, we show that the initial solution has an expected lack/surplus of roughly $\sqrt{n}$ 1-bits compared to any optimal solution even if B is arbitrarily close to $n/2$(Claims 2 & 4). This discrepancy grows linearly when B moves further away from the center $n/2$ (Claim 3).

The second part of the proof consists of deriving bounds on the expected optimization time from these claims.

First, we observe that the problem is equal to unconstrained OneMax (considered as a minimization problem or a maximization problem, respectively), if the cardinality bound is set to the extreme values $B = 0$ or $B = n$. Hence, the well-known $\Theta(n \log n)$ bound [6] carries over to this setting. We assume $B \notin \{0, n\}$ in the following. The upper bound also holds for the uniformly constrained case. We find a feasible solution in time $O(n \log n)$ (Lemma 3) and from there improve it until $B < n$ bits are set to 1. This can be done in an additional phase of $O(n \log n)$ rounds by a Coupon Collector's argument. We proceed in showing that for values of B which are closer to $n/2$ the optimization succeeds much faster.

For the drift analysis it is convenient to use either $|x|_1 = \text{OneMax}(x)$ itself or $|x|_0 = n - |x|_1$ as the potential function, depending on whether the current search string is feasible or not. Our first claim bounds the expected drift with respect to this potential. The results are well-known, we only state them here for completeness. A detailed discussion can be found in [4] and [29].

Claim 1. While the current solution x is feasible, the expected drift is bounded by

$$\frac{|x|_0}{en} \le E\big[|x|_0 - |x'|_0 \;\big|\; |x|_1 < B\big] \le \frac{|x|_0}{n}.$$

Similarly, if x is infeasible, $E[|x|_1 - |x'|_1] = \Theta(|x|_1/n)$.

Next, we give estimates on the second of the above quantities: the distance between the initial number of 1-bits $|x^{(0)}|_1$ and the constraint B. To ease notation, we employ $d_B(x^{(0)}) = |B - |x^{(0)}|_1|$ to denote this distance and note that this is a random variable. Furthermore, we denote by $B_{\text{cen}} = |B - n/2|$ the absolute difference between B and the central value $n/2$, the third quantity.

Claim 2. The expected distance is bounded below by

$$E\Big[d_B(x^{(0)})\Big] = \Omega(\sqrt{n}).$$

Suppose $B \le n/2$, then $|x^{(0)}|_1 \ge (n + \sqrt{n})/2$ is sufficient for $d_B(x^{(0)}) \ge \sqrt{n}/2$. The random variable $|x^{(0)}|_1$ is the sum of n i.i.d. Bernoulli trials and thus has expected value $n/2$ and standard deviation $\sqrt{n}/2$. An application of Lemma 6 in [24] now gives

$$P\Big[|x^{(0)}|_1 \ge \frac{n}{2} + \frac{\sqrt{n}}{2}\Big] = \Omega(1).$$

If $B > n/2$, we apply the same argument to the random variable $|x^{(0)}|_0$ (having the same distribution) and the event $|x^{(0)}|_0 \ge (n + \sqrt{n})/2$. By Lemma 2 we obtain in both cases

$$E\Big[d_B(x^{(0)})\Big] \ge \Omega(1) \cdot \frac{\sqrt{n}}{2} = \Omega(\sqrt{n}).$$

Claim 3. The expected distance is bounded below by

$$E\Big[d_B(x^{(0)})\Big] \ge \frac{B_{\text{cen}}}{2}.$$

The median of the random variable $|x^{(0)}|_1$ is $n/2$. Hence, if $B \le n/2$, with probability $P[|x^{(0)}|_1 \ge n/2] \ge 1/2$ the distance is at least B_{cen}. The same holds if $B > n/2$ as also $P[|x^{(0)}|_1 \le n/2] \ge 1/2$. The lower bound again is due to Lemma 2.

Claim 4. The expected distance is bounded above by

$$E\Big[d_B(x^{(0)})\Big] \le B_{\text{cen}} + \frac{e}{4\pi}\sqrt{n}.$$

The Triangle Inequality yields $d_B(x^{(0)}) \le B_{\text{cen}} + ||x^{(0)}|_1 - n/2|$. By the monotonicity and linearity of expectations we deduce

$$\begin{aligned} E\Big[d_B(x^{(0)})\Big] &\le E\Big[B_{\text{cen}} + \Big||x^{(0)}|_1 - \frac{n}{2}\Big|\Big]\\ &= B_{\text{cen}} + E\Big[\Big||x^{(0)}|_1 - \frac{n}{2}\Big|\Big].\end{aligned}$$

The latter expected value is known as the *mean deviation* of a binomially distributed random variable. For the special case of success probability $1/2$ the mean deviation equals $\binom{n}{\lceil n/2\rceil}\lceil n/2\rceil 2^{-n}$, cf. e.g. [7]. Applying Stirling's approximations of the factorial, we obtain

$$\begin{aligned} E\Big[\Big||x^{(0)}|_1 - \frac{n}{2}\Big|\Big] &= \frac{n}{2^{n+1}}\binom{n}{\frac{n}{2}} = \frac{n}{2^{n+1}} \cdot \frac{n!}{\left(\frac{n}{2}!\right)^2}\\ &\le \frac{n}{2^{n+1}} \cdot \frac{e\sqrt{n}\left(\frac{n}{e}\right)^n}{\left(\sqrt{2\pi}\sqrt{\frac{n}{2}}\left(\frac{n}{2e}\right)^{\frac{n}{2}}\right)^2} = \frac{e}{2\pi}\sqrt{n}.\end{aligned}$$

Claims 2, 3 and 4 together show that if $B_{\text{cen}} < \sqrt{n}$, the initial solution has distance $E[d_B(x^{(0)})] = \Theta(\sqrt{n})$; otherwise, it is in $\Theta(B_{\text{cen}})$.

The last claim states a useful technical property of the first feasible solution found during the optimization, as described in Lemma 3.

Claim 5. If the initial solution was infeasible, with probability superpolynomially close to 1 the first feasible solution sampled by the (1+1) EA has Hamming weight at least $B - \ln n$.

Consider the iteration in which the optimization process enters the feasible region. In order to jump from more than B bits set to 1 to less than $B - \ln n$, at least $\ln n$ bits must flip at once. We get the following bound on the probability,

$$P\big[\,|x'|_1 < B - \ln n \;\big|\; |x|_1 > B\big] \leq \sum_{i=\ln n}^{n} \binom{n}{i} \frac{1}{n^i}$$
$$\leq n \binom{n}{\ln n} \frac{1}{n^{\ln n}} \leq n \left(\frac{e}{\ln n}\right)^{\ln n} = \frac{1}{n^{\ln \ln n - 2}}.$$

In the remainder of this proof we infer bounds on the expected optimization time from the claims above. We commence with proving an universal lower bound. Claim 1 states that the drift during the whole optimization is at most 1, regardless of feasibility. We recall that T is the random variable denoting the number of rounds the (1+1) EA needs to sample an optimal solution for the first time. Its expected value $E[T \,|\, d_B(x^{(0)})]$ conditional on the distance of the initial solution to the bound B is again a random variable. Suppose this distance $d_B(x^{(0)})$ is equal to some natural number $0 \leq d \leq n$. The Additive Drift Theorem for lower bounds [11] asserts, for all such d,

$$E\Big[T \;\Big|\; d_B(x^{(0)}) = d\Big] \geq d.$$

Utilizing Claim 2, we bound the expectation of the *derived* variable and, in turn, the expected optimization time,

$$E[T] = E\Big[E\Big[T \;\Big|\; d_B(x^{(0)})\Big]\Big] \geq E\Big[d_B(x^{(0)})\Big] = \Omega(\sqrt{n}).$$

Note that this bound holds for any uniform constraint B.

According to Chernoff bounds, the initial solution has at least $n/3$ and at most $2n/3$ bits set to 1 with probability $1 - 2^{-\Omega(n)}$. We recall that the upper bound of $\mathrm{O}(n \log n)$ rounds holds for all values of B. Hence, conditioning on $x^{(0)}$ to contain a linear number of both 1s and 0s affects the expected run time only by a sub-constant number of iterations. We omit this condition in the notation below.

Suppose $B_{\text{cen}} < \sqrt{n}$. While the currently best search point is infeasible, the number of 1-bits cannot increase. However, undershooting the target cardinality bound B by more than $\ln n$ is also unlikely (Claim 5). Conversely, the number of 1-bits cannot decrease while the search point is feasible. In summary, we can assume that the maintained solution x observes $n/3 \leq |x|_1 \leq 2n/3$ during the whole optimization. By Claim 1, the expected drift is at least $1/3e$. The Additive Drift Theorem for upper bounds [11] now yields

$$E\Big[T \;\Big|\; d_B(x^{(0)})\Big] \leq 3e \cdot d_B(x^{(0)}).$$

Applying Claim 4 and the same technique as above, we obtain

$$E[T] = E\Big[E\Big[T \;\Big|\; d_B(x^{(0)})\Big]\Big] \leq 3e \cdot E\Big[d_B(x^{(0)})\Big]$$
$$\leq 3e \left(B_{\text{cen}} + \frac{e}{2\pi}\sqrt{n}\right) = \mathrm{O}(\sqrt{n}).$$

In the case of B_{cen} to be between $\sqrt{n}$ and $n/4$, the expected optimization time is in $\Theta(B_{\text{cen}})$. The argument is analogue to above involving Claim 3 (instead of Claim 2) as well as Claim 4. The main observation is that the drift can still assumed to be a constant in this range of B. Note that now $\mathrm{O}(B_{\text{cen}} + \sqrt{n}) = \mathrm{O}(B_{\text{cen}})$.

Finally, we turn the investigation to the case wherethe distance B_{cen} is larger than $n/4$. The main difference is that the expected drift can now become sub-constant during the optimization. We use a multiplicative drift argument to handle this issue.

First, we treat the case $B \geq 3n/4$. This implies $B_{\min} = \min\{B, n - B\} = n - B$. Furthermore, by the Chernoff argument shown above, the initial solution is feasible (with probability exponentially close to 1). We employ the number of 0-bits as the potential, which can be at most n. In order to optimize OneMax, the (1+1) EA has to generate an offspring with potential $B_{\min}$. By Claim 1 the expected drift is at least $|x|_0/en$. The Multiplicative Drift Theorem for upper bounds [4] now yields

$$E[T] \leq en \left(\ln\left(\frac{n}{B_{\min}}\right) + 1\right) = \mathrm{O}\left(n \log\left(\frac{n}{B_{\min}}\right)\right).$$

Regarding the lower bound, we can assume that the initial solution is not only feasible but has at least $B_{\min} + n/12$ bits set to 0. This implies that the number of 0-bits cannot increase in the optimization. We only measure the time until this number is lower than $B_{\min} + \ln n$. Suppose the current potential is $|x|_0 = k$. Then, the expected drift is at most $k/n =: \delta k$ (Claim 1). Additionally, large jumps are unlikely. More formally, in order to have a progress of at least $k/2 =: \beta k$, between $k/2$ and k bits must flip simultaneously. The probability for such a mutation is at most

$$\sum_{i=k/2}^{k} \binom{n}{i} \frac{1}{n^i} \leq \sum_{i=k/2}^{k} \left(\frac{e}{i}\right)^i \leq \frac{k}{2}\left(\frac{2e}{k}\right)^{\frac{k}{2}}.$$

Since $k \geq B_{\min} + \ln n > \ln n$, we obtain

$$P\left[|x|_0 - |x'|_0 \geq \frac{k}{2} \;\middle|\; |x|_0 = k\right] < \frac{\ln n}{2}\left(\frac{2e}{\ln n}\right)^{\frac{\ln n}{2}}$$
$$\leq \frac{1}{2n \ln k} = \frac{\beta\delta}{\ln k}$$

for n sufficiently large. Hence, the conditions for the Multiplicative Drift Theorem for lower bounds [29] with parameters $\delta = 1/n$ and $\beta = 1/2$ are satisfied and we obtain

$$E[T] \geq \frac{1}{\delta}\,\ln\left(\frac{B_{\min} + n/12}{B_{\min} + \ln n}\right)\frac{1-\beta}{1+\beta}$$
$$\geq \frac{n}{3}\ln\left(\frac{n/12}{B_{\min} + \ln n}\right) \geq \frac{n}{6}\ln\left(\frac{n}{12\,B_{\min}}\right)$$
$$= \Omega\left(n \log\left(\frac{n}{B_{\min}}\right)\right).$$

The proofs of the run time bounds in case of $B \leq n/4$ are similar, but somehow simpler. Note that now $B_{\min} = B$.

For the analysis we invert the roles of 0- and 1-bits. With probability exponentially close to 1 the initial solution is *infeasible* and has a linear surplus of *1-bits*. A reduction below $B_{\min} + \ln n$ is necessary to optimize the bit string. By the same arguments as above this needs an expected number of $\Omega(n \log(n/B))$ iterations. To derive an upper bound, we argue that the search for a feasible solution, which takes time $O(n \log(n/B))$ in expectation (Lemma 3), dominates the run time. Once the (1+1) EA enters the feasible region, by Claim 5 we only need to collect $\ln n$ additional 1-bits with probability superpolynomially close to 1. Since $B \leq n/4$, the currently best solution x yields $|x|_0 \geq 3n/4$. Thus, we again observe a constant drift. Summarizing the two phases yields

$$E[T] = O\Big(n \log\Big(\frac{n}{B}\Big)\Big) + O(\log n) = O\Big(n \log\Big(\frac{n}{B_{\min}}\Big)\Big). \quad \square$$

3.2 Linear Functions

We move to the general case of linear objective functions under uniform constraint. The weights $w_i \geq 1$ are now chosen arbitrarily, whereas every b_i still is equal to 1. Contrary to our results on OneMax, the introduction of constraints increases the optimization time of linear functions in general. The reason is that during the optimization the increase of 1-bits stalls at the cardinality bound B. From there, progress is only possible by swapping a 1-bit to a position with larger weight, currently set to 0. This requires a simultaneous flip of both bits.

Theorem 5. *There is a linear function f and a bound B such that the optimization time of the (1+1) EA on f under uniform constraint B is in $\Omega(n^2)$, not only in expectation but even with high probability.*[1]

Proof. Let $\varepsilon > 0$ be an arbitrary positive quantity, possibly even dependent on n. We set the bound $B = 3n/4$ and define function f as

$$f(x) = \sum_{i=1}^{B} (1+\varepsilon) x_i + \sum_{j=B+1}^{n} x_j.$$

The slight weight increase in the first B bits results in f to having its *unique* global optimum at $x^* = 1^{3n/4}0^{n/4}$, contrary to the $\binom{n}{B}$ optima in the case of OneMax. The main idea of this proof is to show that during the optimization of f under constraint B the (1+1) EA w.h.p. samples a point with constant Hamming distance d_H from the optimum and with exactly B 1-bits. Then, the only way to reach the optimum is to exchange a 0 in the first $3n/4$ bit positions, the *first block*, with a 1 in the last $n/4$ bits, the *second block*. This event has a waiting time in $\Omega(n^2)$.

First, we prove that the (1+1) EA (again w.h.p.), before finding x^*, either samples a search point with Hamming distance between 4 and 8 from the optimum or runs for $\Omega(n^2)$ iteration regardless. We then show that, given such a feasible solution with constant Hamming distance, the algorithm finds another bit string with exactly B 1-bits prior to the optimal one. Finally, a union bound over the polynomially small error probabilities for these events implies the theorem.

[1]We use the term *with high probability* (w.h.p.) for a success probability of at least $1 - n^{-c}$ for some constant $c > 0$.

By Chernoff bounds the initial solution has no more than $2n/3$ bits set to 1 with probability exponentially close to 1. Thus, we observe a linear Hamming distance from x^*. In order to maximize function f the (1+1) EA must decrease this distance below any positive constant. We argue that the algorithm does not jump directly from an individual with distance greater than 8 to one with distance less than 4. To this end, let $d > 8$ be the number of wrongly set bits of the current search point x. We pessimistically assume that every mutation decreasing the distance is accepted. For this mutation at most 3 of these d bits are allowed to not flip at once. The probability for this event is

$$P\big[d_H(x', x^*) < 4 \mid d_H(x, x^*) = d\big] \leq \sum_{i=0}^{3} \binom{d}{d-i} \frac{1}{n^{d-i}}$$
$$\leq 4\binom{d}{3} \frac{1}{n^{d-3}} \leq \frac{d^3}{n^{d-3}} = O\Big(\frac{1}{n^6}\Big).$$

The last estimate is due to the observation that the upper bound is maximal when $d = 9$. Therefore, for a suitable constant $c > 0$ this jump does not occur in the first cn^2 steps of the optimization with probability at least $1 - 1/n^4$.

We now assume that we are given a feasible solution x with $4 \leq d_H(x, x^*) \leq 8$ and continue the analysis from this point on. If x has exactly B 1-bits, the theorem follows immediately. The reason is as follows. Due to the Hamming distance x can have at most 8 0-bits in the first block and at most 8 1-bits in the second one. Every mutation must flip at least one of these misplaced 0s and 1s in B simultaneously to improve on the fitness value. The probability for this to happen is at most $8^2/n^2$.

What is left is the case where x has Hamming distance between 4 and 8, and strictly less than B 1-bits. However, again due to the distance, $|x|_1 \geq B - 8$ must hold. Consider a run of the (1+1) EA for $t^* = \ln n$ steps. Employing drift analysis, we show that during this phase the current best search point collects B 1s in total but still does not reach x^* w.h.p. A standard argument provides a lower bound of $E[|x'|_1 - |x|_1 \mid |x|_1 < B] \geq 1/4e =: \delta$ on the expected drift since x has more than $\overline{B} = n/4$ bits set to 0 and flipping any of them is accepted as a fitness increase. Furthermore, observe that no mutation incrementing the number of 1s by more than $8 =: s$ is accepted as this would violate the constraint. A tail bound for the Additive Drift Theorem [15] with parameters $\delta = 1/4e$ and $s = 8$ yields that the probability of the (1+1) EA to remove the surplus of 0s within t^* rounds is at least

$$1 - \exp\Big(-\frac{t^* \delta^2}{8s^2}\Big) = 1 - \exp(-\Omega(\log n)) = 1 - \frac{1}{n^{\Omega(1)}}.$$

We are allowed to assume that the Hamming distance to the optimum does not increase beyond 8 during these t^* steps as otherwise the argument presented above still gives a quadratic lower bound on the run time. On the other hand, in order to reach x^* in this phase all $d \geq 4$ wrongly set bits would have to flip at least once. It is left to prove that this does not happen with high probability: A specific bit position does not flip during t^* rounds with probability $(1 - 1/n)^{t^*}$. Hence, all d bits flip during this phase with probability $(1 - (1 - 1/n)^{t^*})^d$. Therefore, the (1+1) EA does not sample the optimum during the t^* rounds with probability at least

$$1-\left(1-\left(1-\frac{1}{n}\right)^{t^*}\right)^d \geq 1-\left(\frac{t^*}{n}\right)^d \geq 1-\left(\frac{\ln n}{n}\right)^4 \geq 1-\frac{1}{n^2}.$$

The estimate is due to Bernoulli's Inequality and the observation that $\ln n \leq \sqrt{n}$ for n large enough. □

Theorem 6. *For arbitrary values of B, the expected optimization time of the (1+1) EA on any linear function under uniform constraint B is in $\mathrm{O}(n^2 \log(B\, w_{max}))$.*

Proof. We start the analysis with a feasible solution due to Lemma 3. This implies that the (1+1) EA will never sample an infeasible solution from this point on. W.l.o.g. the weights of function f are in descending order starting with the left-most bit, i.e. $w_{\max} = w_1 \geq w_2 \geq \cdots \geq w_n$. f under uniform constraint B has a maximum objective value of $f_{\max} = \sum_{i=1}^{B} w_i$. For a search point x we assign the potential function $g(x) = f_{\max} - f(x)$. This potential is non-negative and attains its minimum value 0 just in case $f(x)$ is optimal.

We again refer to the first B bits as the first block and the remaining $n - B$ bits as the second block and recall that $\overline{x_i}$ denote the negation of the bit at position i. We define two auxiliary functions

$$\mathrm{loss}(x) = \sum_{i=1}^{B} w_i \overline{x_i}; \quad \mathrm{surplus}(x) = \sum_{j=B+1}^{n} w_j x_j.$$

Thus, $\mathrm{loss}(x)$ measures the total weights of the missing positions in the first block, while $\mathrm{surplus}(x)$ is the sum of weights of the superfluous ones in the second block. We can reformulate the potential as

$$g(x) = \mathrm{loss}(x) - \mathrm{surplus}(x).$$

Suppose the current best solution x is non-optimal, let $k \geq 1$ denote the number of 0-bits in the first block of x. We pessimistically assume that $|x|_1$ reached the cardinality bound B. In this case the expected drift with respect to the above potential is minimal, since for $|x|_1 < B$ already a mutation flipping a single 0-bit could improve on the fitness value. Hence, there are exactly k corresponding 1-bits in the second block. Let $\mathcal{A}_{1,2}$ denote the event that one 0 in the first block, one 1 in the second and no other position flips in this round. By the law of total expectation, we bound the expected drift of $g(x)$ with the conditional drift under $\mathcal{A}_{1,2}$. Any of the k 0s in the first block are equally likely to flip and together they make up for the whole value of $\mathrm{loss}(x)$. Thus, the average weight increase by flipping one of them is $\mathrm{loss}(x)/k$. An analogue argument regarding $\mathrm{surplus}(x)$ applies to the weight decrease by flipping one of the k 1s in the second block. We sum these by Lemma 2 on the expected drift of g and the event $\mathcal{A}_{1,2}$:

$$E[g(x) - g(x')] \geq E[g(x) - g(x') \mid \mathcal{A}_{1,2}]\, P[\mathcal{A}_{1,2}]$$
$$\geq \left(\frac{\mathrm{loss}(x)}{k} - \frac{\mathrm{surplus}(x)}{k}\right)\frac{k^2}{en^2} = g(x)\frac{k}{en^2} \geq \frac{g(x)}{en^2}.$$

We recall that $g(x)$ on feasible solutions is never larger than $f_{\max} = \sum_{i=1}^{B} w_i$. The Multiplicative Drift Theorem [4] implies

$$E[T] \leq en^2\,(\ln(f_{\max}) + 1) \leq en^2\,(\ln(B\, w_{\max}) + 1). \quad \square$$

Consider the BinVal-function defined by $\mathrm{BinVal}(x) = \sum_{i=1}^{n} 2^{n-i} x_i$. It serves as one extreme example of a linear function where any weight is strictly larger than the sum of all smaller weights. Hence, we can assume $w_{\max} \leq 2^n$ and Theorem 6 implies a worst-case expected optimization time in $\mathrm{O}(n^3)$ for all linear functions and arbitrary uniform constraint B. However, we suspect the log-factor appearing in the above bound to be an artifact of the analysis and consequently conjecture that every linear function can be optimized in time $\mathrm{O}(n^2)$. As an example we return to the class of functions used in the proof of Theorem 5.

Theorem 7. *Let $0 < \varepsilon < 1/n$ be a positive real. Then, the expected optimization time of the (1+1) EA on the function $f(x) = \sum_{i=1}^{B}(1+\varepsilon)\, x_i + \sum_{j=B+1}^{n} x_j$ under uniform constraint B is in $\mathrm{O}(n^2)$.*

Proof. Due to Lemma 3 the assumption of a feasible search point x as start for our investigation does not affect the asymptotic run time. The key observation of this proof is that any mutation of a feasible solution that reduces the number of 1-bits is rejected, implying a process similar to OneMax. In order for the offspring to be accepted, the fitness cannot be worse than the parents. Hence, any 1-bit deleted from the last $n - B$ bits (the second block) must be balanced by the gain of an additional 1-bit in the first B bits (the first block). Note that due to $\varepsilon < 1/n$ no fewer number of 1-bits suffice. Conversely, if a mutation reduces the number of 1-bits in the first block, one even needs a strict increase of 1-bits in the second block to compensate for that due to $\varepsilon > 0$.

The first statement of Theorem 4 asserts that in expectation within some $\mathrm{O}(n \log n)$ iterations we sample a string x with $|x|_1 = B$. We also stay at the cardinality bound until the optimization is finished. We define the potential of a partial solution as the number of missing bits in the first block or, more formally, $g(x) = \sum_{i=1}^{B} \overline{x_i}$. Suppose the current potential is $g(x) = k$. Since we have reached the bound, the second block contains exactly k bits set to 1. Focusing on mutations which flip a single 0-bit in the first block and a single 1-bit in the second block, we bound the expected drift by

$$E[g(x) - g(x') \mid g(x) = k] \geq \frac{k^2}{n^2}\left(1 - \frac{1}{n}\right)^{n-2} \geq \frac{k^2}{en^2} =: h(k).$$

Let T_1 be the random variable denoting the number of iterations until the (1+1) EA reaches the optimum $x^* = 1^B 0^{n-B}$ starting from a solution with B 1-bits. The Variable Drift Theorem (Theorem 1, [13, 25]) applied to function h with $k_0 \leq B$ and $k_{\min} = 1$ yields a bound on the expected value

$$E[T_1] \leq \frac{k_{\min}}{h(k_{\min})} + \int_{k_{\min}}^{k_0} \frac{1}{h(k)}\,\mathrm{d}k$$
$$= en^2\left(1 + \int_1^B \frac{1}{k^2}\,\mathrm{d}k\right) \leq 2en^2.$$

Together with the bound on finding a solution with exactly B 1-bits, this implies the theorem. □

3.3 LeadingOnes

Another well-studied but *non*-linear pseudo-Boolean function, admits a similar run time bound as in Theorem 6. LeadingOnes is defined as

$$\text{LeadingOnes}(x) = \sum_{i=1}^{n} \prod_{j=1}^{i} x_j$$

and counts the number of consecutive 1-bits starting from the left-most bit. Under uniform constraint B, the unique optimum is the string $1^B 0^{n-B}$.

Theorem 8. *For arbitrary values of B, the expected optimization time of the (1+1) EA on* LeadingOnes *under uniform constraint B is in* $\mathrm{O}(n^2 \log B)$.

Proof. Again we can assume the optimization to start with a feasible solution without affecting the asymptotic run time due to Lemma 3. For a given search point x we assign the potential function $g(x)$ by

$$g(x) = B - \text{LeadingOnes}(x).$$

It suffices to bound the expected drift of $g(x)$ in the worst case, where the current solution x is non-optimal and has exactly B bits set to 1. Suppose x has potential $g(x) = k$, then the structure of x is as follows. Starting with the left-most bit there is a consecutive substring of $B - k$ 1-bits followed by a single 0, the remaining $n - B + k - 1$ bits form a substring in which exactly k positions are set to 1. In order to reduce the current potential, it suffices that the prominent first 0 as well as exactly one 1 among the later bits flip. This results in an expected drift of at least

$$E[g(x) - g(x') \mid g(x) = k] \geq \frac{1}{n}\frac{k}{n}\left(1 - \frac{1}{n}\right)^{n-2} \geq \frac{k}{en^2}.$$

Due to the Multiplicative Drift Theorem [4] we derive the bound for the expected time to reduce the potential of any feasible solution to 0

$$E[T] \leq en^2(\ln(\max g(x)) + 1) = en^2(\ln B + 1). \qquad \square$$

4. LINEAR CONSTRAINT

We now investigate linear functions under linear constraint, i.e. arbitrary $b_i > 0$ for the constraint function. The resulting optimization problem for bit strings is capable of encoding the NP-complete KnapSack problem, cf. [14]. Thus, it is not surprising that the (1+1) EA needs exponential run time already on the restricted case of OneMax as the objective function. Furthermore, it is well known that there are trap-like problem instances fitting the general knapsack formulation that can not be solved efficiently by simple evolutionary algorithms [10] working with the problem formulation considered in this paper. The reason for this is that they get trapped in a local optimum which has a large Hamming distance to the globally optimal solution. The class instances investigated in [10] consists of $n - 1$ items having a weight and profit of 1 and one item having a large weight and profit. We show that even OneMax under a particular linear constraint can not be optimized efficiently by the (1+1) EA.

Theorem 9. *There is a linear function $b(x)$ and a bound B such that the optimization time of the (1+1) EA on* OneMax *under linear constraint $b(x) \leq B$ is in $2^{\Omega(n)}$, not only in expectation but even with overwhelming probability.*[2]

Proof. We define the constraint function $b(x)$ as

$$b(x) = \sum_{i=1}^{2n/3} n x_i + \sum_{i=2n/3+1}^{n} (n+1) x_i$$

together with the bound $B = 2n^2/3$. Therefore, we ensure that every bit string x with $|x|_1 < 2n/3$ is feasible while the ones with $|x|_1 \geq 2n/3$ are infeasible. The *sole exception* is the optimal solution $x^* = 1^{2n/3}0^{n/3}$. In other words, the collection of strings with exactly $2n/3 - 1$ bits set to 1 form a large plateau of equal but non-optimal fitness. We condition the following analysis on the initial solution being feasible, which happens with overwhelming probability due to Chernoff bounds. After that, the (1+1) EA never adopts an infeasible search point as the current best.

We prove an *unbiasedness* property [19] of the underlying random process. Informally, as long as the optimum has not been found, the probability of a bit string to be sampled in round t depends only on the number of the bits set to 1, not on their position. In order to state this more formally, we first need some additional notation highlighting the effect of selection in the optimization. Let $(X^{(t)})_{t\geq 0}$ be the series of random variables denoting the search points adopted *after* the selection in round t. For $t > 0$ $Y^{(t)}$ denotes the offspring (of individual $X^{(t-1)}$) created in round t *before* any selection takes place, whereas $Y^{(0)} = X^{(0)}$ is the initial solution. For a permutation $\pi\colon \{1, \dots n\} \to \{1, \dots n\}$ and bit string $x \in \{0,1\}^n$ let $\pi(x) = x_{\pi(1)}x_{\pi(2)} \dots x_{\pi(n)}$ be the string obtained from x by deranging its positions according to π.

While x^* has not yet been found the search behaves like the (1+1) EA on unconstrained OneMax with a rejection of solutions with Hamming weight at least $2n/3$. We claim that for any $t > 0$, permutation π and bit string $y \in \{0,1\}^n$

$$P[Y^{(t)} = y \mid X^{(t-1)} \neq x^*] = P[Y^{(t)} = \pi(y) \mid X^{(t-1)} \neq x^*].$$

We prove this by induction over t. Lehre and Witt have characterized the mutation operator of the (1+1) EA as an *unary, unbiased variation operator* [19]; in particular, it is invariant under permutation, that is

$$P[Y^{(t)} = y \mid X^{(t-1)} = x] = P[Y^{(t)} = \pi(y) \mid X^{(t-1)} = \pi(x)].$$

For the initial $X^{(0)}$ every bit string is equally likely to be chosen satisfying the claim.

The search point $X^{(t-1)}$ is equal to the offspring $Y^{(t^*)}$ of the round $t^* < t$ in which it was selected. By the induction hypothesis the claim holds for t^*. The selection itself is also unbiased as the objective function OneMax (with the additional rejection rule) is invariant under bit permutation. This implies that we can apply the law of total probability to express $P[Y^{(t)} = y \mid X^{(t-1)} \neq x^*]$ as a sum over all conditional probabilities that the current best (feasible, non-optimal) solution is x and its offspring is y. We conclude

[2]We use the term *with overwhelming probability* for a success probability of at least $1 - 2^{-cn}$ for some constant $c > 0$.

$$
\begin{aligned}
&P[Y^{(t)} = y \mid X^{(t-1)} \neq x^*] \\
&= \sum_{|x|_1 < \frac{2}{3}n} P[Y^{(t)} = y \mid X^{(t-1)} = x]\, P[X^{(t-1)} = x] \\
&= \sum_{|x|_1 < \frac{2}{3}n} P[Y^{(t)} = \pi(y) \mid X^{(t-1)} = \pi(x)]\, P[X^{(t-1)} = \pi(x)] \\
&= P[Y^{(t)} = \pi(y) \mid X^{(t-1)} \neq x^*].
\end{aligned}
$$

There are $\binom{n}{n/3}$ bit strings with weight $2n/3$, but only one of them is optimal. The unbiasedness regarding permutation implies that for any t the probability of finding x^* in round t is at most

$$
P[Y^{(t)} = x^* \mid X^{(t-1)} \neq x^*] \leq \binom{n}{n/3}^{-1} \leq 3^{-\frac{n}{3}},
$$

which yields an exponential waiting time. Moreover, a union bound shows that for a suitably small constant $c > 0$ the probability of finding the optimum within the first 2^{cn} iterations is exponentially small. □

5. CONCLUSION

Studying the run time behavior of linear functions has provided many new tools for analyzing evolutionary computing techniques and set the basis for run time studies for more complex problems. With this paper we have contributed to the area of run time analysis of evolutionary computing by studying classes of linear functions under a given linear constraint. This is equivalent to special classes of the well-known knapsack problem. Central to the area of run time analysis for linear functions is the function OneMax. In our study we have focused on problem classes where the objective function or the constraint function is given by OneMax.

Our theoretical investigations show that the (1+1) EA can handle uniform constraints efficiently, but fails for more general constraints even on OneMax. The constraint handling we employed directs the search within the infeasible region towards the feasible region by adding a penalty dependent on the distance to the constraint. However, the search within the feasible region is not guided by any knowledge about the constraint. Therefore, it is interesting to investigate whether additional information can help direct the search such that OneMax with non-uniform constraint can be handled efficiently.

Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 618091 (SAGE) and from the Australian Research Council under grant agreements DP140103400 and DP160102401.

6. REFERENCES

[1] A. Auger and B. Doerr. *Theory of Randomized Search Heuristics: Foundations and Recent Developments.* World Scientific Publishing, 2011.

[2] R. Beier. *Probabilistic Analysis of Discrete Optimization Problems.* PhD thesis, Universität des Saarlandes, 2005.

[3] D. Dang and P. K. Lehre. Simplified Runtime Analysis of Estimation of Distribution Algorithms. In *Proc. of GECCO'15*, pages 513–518, 2015.

[4] B. Doerr, D. Johannsen, and C. Winzen. Multiplicative Drift Analysis. *Algorithmica*, 64:673–697, 2012.

[5] S. Droste. Analysis of the (1+1) EA for a Noisy OneMax. In *Proc. of GECCO'04*, pages 1088–1099, 2004.

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

[7] J. S. Frame. Mean Deviation of the Binomial Distribution. *The American Mathematical Monthly*, 52:377–379, 1945.

[8] C. Gießen and T. Kötzing. Robustness of Populations in Stochastic Environments. *Algorithmica*, 75:462–489, 2016.

[9] M. Harman. Software Engineering Meets Evolutionary Computation. *IEEE Computer*, 44:31–39, 2011.

[10] J. He, B. Mitavskiy, and Y. Zhou. A theoretical assessment of solution quality in evolutionary algorithms for the knapsack problem. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, Beijing, China, July 6-11, 2014*, pages 141–148. IEEE, 2014.

[11] J. He and X. Yao. A study of drift analysis for estimating computation time of evolutionary algorithms. *Natural Computing*, 3:21–35, 2004.

[12] T. Jansen. *Analyzing Evolutionary Algorithms - The Computer Science Perspective.* Natural Computing Series. Springer, 2013.

[13] D. Johannsen. *Random Combinatorial Structures and Randomized Search Heuristics.* PhD thesis, Universität des Saarlandes, 2010.

[14] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems.* Springer, 2004.

[15] T. Kötzing. Concentration of First Hitting Times Under Additive Drift. *Algorithmica*, 75:490–506, 2016.

[16] T. Kötzing, A. Lissovoi, and C. Witt. (1+1) EA on Generalized Dynamic OneMax. In *Proc. of FOGA'15*, pages 40–51, 2015.

[17] T. Kötzing, F. Neumann, D. Sudholt, and M. Wagner. Simple Max-Min Ant Systems and the Optimization of Linear Pseudo-Boolean Functions. In *Proc. of FOGA'11*, pages 209–218, 2011.

[18] C. Le Goues, T. Nguyen, S. Forrest, and W. Weimer. GenProg: A Generic Method for Automatic Software Repair. *IEEE Transaction on Software Engineering*, 38:54–72, 2012.

[19] P. K. Lehre and C. Witt. Black-Box Search by Unbiased Variation. *Algorithmica*, 64:623–642, 2012.

[20] X. Li, M. R. Bonyadi, Z. Michalewicz, and L. Barone. Solving a Real-world Wheat Blending Problem Using a Hybrid Evolutionary Algorithm. In *Proc. of CEC'13*, pages 2665–2671, 2013.

[21] D. Lückehe, M. Wagner, and O. Kramer. On Evolutionary Approaches to Wind Turbine Placement with Geo-Constraints. In *Proc. of GECCO'15*, pages 1223–1230, 2015.

[22] H. Mühlenbein. How Genetic Algorithms Really Work: Mutation and Hillclimbing. In *Proc. of PPSN'92*, pages 15–26, 1992.

[23] F. Neumann and C. Witt. *Bioinspired Computation in Combinatorial Optimization:Algorithms and Their Computational Complexity*. Springer, 2010.

[24] P. S. Oliveto and C. Witt. Improved Time Complexity Analysis of the Simple Genetic Algorithm. *Theoretical Computer Science*, 605:21–41, 2015.

[25] J. E. Rowe and D. Sudholt. The Choice of the Offspring Population Size in the (1,λ) EA. In *Proc. of GECCO'12*, pages 1349–1356, 2012.

[26] D. A. Spielman and S. Teng. Smoothed analysis: an attempt to explain the behavior of algorithms in practice. *Communications of the ACM*, 52:76–84, 2009.

[27] C. S. Stokes, A. R. Simpson, and H. R. Maier. A Computational Software Tool for the Minimization of Costs and Greenhouse Gas Emissions Associated with Water Distribution Systems. *Environmental Modelling and Software*, 69:452–467, 2015.

[28] D. Sudholt and C. Witt. Runtime Analysis of a Binary Particle Swarm Optimizer. *Theoretical Computer Science*, 411:2084–2100, 2010.

[29] C. Witt. Tight Bounds on the Optimization Time of a Randomized Search Heuristic on Linear Functions. *Combinatorics, Probability and Computing*, 22:294–318, 2013.

[30] F. Zheng, A. R. Simpson, and A. C. Zecchin. Improving the Efficiency of Multi-objective Evolutionary Algorithms Through Decomposition: An Application to Water Distribution Network Design. *Environmental Modelling and Software*, 69:240–252, 2015.

[31] Y. Zhou and J. He. A Runtime Analysis of Evolutionary Algorithms for Constrained Optimization Problems. *IEEE Transactions on Evolutionary Computation*, 11:608–619, 2007.