

# Black-Box Search by Unbiased Variation

Per Kristian Lehre · Carsten Witt

Received: 21 December 2010 / Accepted: 31 January 2012 / Published online: 9 February 2012  
© Springer Science+Business Media, LLC 2012

**Abstract** The complexity theory for black-box algorithms, introduced by Droste, Jansen, and Wegener (Theory Comput. Syst. 39:525–544, 2006), describes common limits on the efficiency of a broad class of randomised search heuristics. There is an obvious trade-off between the generality of the black-box model and the strength of the bounds that can be proven in such a model. In particular, the original black-box model provides for well-known benchmark problems relatively small lower bounds, which seem unrealistic in certain cases and are typically not met by popular search heuristics.

In this paper, we introduce a more restricted black-box model for optimisation of pseudo-Boolean functions which we claim captures the working principles of many randomised search heuristics including simulated annealing, evolutionary algorithms, randomised local search, and others. The key concept worked out is an unbiased variation operator. Considering this class of algorithms, significantly better lower bounds on the black-box complexity are proved, amongst them an  $\Omega(n \log n)$  bound for functions with unique optimum. Moreover, a simple unimodal function and plateau functions are considered. We show that a simple  $(1 + 1)$  EA is able to match the runtime bounds in several cases.

**Keywords** Runtime analysis · Black-box complexity

---

This work is based on earlier work in [21].

P.K. Lehre was supported by EPSRC under grant no. EP/D052785/1, and Deutsche Forschungsgemeinschaft (DFG) under grant no. WI 3552/1-1.

C. Witt was supported by Deutsche Forschungsgemeinschaft (DFG) under grant no. WI 3552/1-1.

P.K. Lehre (✉)

School of Computer Science, University of Nottingham, Jubilee Campus, Wollaton Road,  
Nottingham NG8 1BB, UK  
e-mail: [PerKristian.Lehre@nottingham.ac.uk](mailto:PerKristian.Lehre@nottingham.ac.uk)

C. Witt

Technical University of Denmark, Kgs. Lyngby, Denmark  
e-mail: [cwf@imm.dtu.dk](mailto:cwf@imm.dtu.dk)

## 1 Introduction

The theory of randomised search heuristics has advanced significantly over the last years. In particular, there exist now rigorous results giving the runtime of canonical search heuristics, like the  $(1 + 1)$  EA, on combinatorial optimisation problems [25]. Ideally, these theoretical advances will guide practitioners in their application of search heuristics. However, it is still unclear to what degree the theoretical results that have been obtained for the canonical search heuristics can inform the usage of the numerous, and often more complex, search heuristics that are applied in practice. While there is an ongoing effort in extending runtime analysis to the more complex search heuristics, this often requires development of new mathematical techniques.

To advance the theoretical understanding of search heuristics, it would be desirable to develop a computational complexity theory of search heuristics. The basis of such a theory would be a computational model that captures the inherent limitations of search heuristics. The results would be a classification of problems according to the time required to solve the problems in the model. Such a theory has already been introduced for local search problems [17]. The goal in *local search* problems is to find any solution that is locally optimal with regards to the cost function and a neighbourhood structure over the solution set. Here, we are interested in *global search* problems, i.e. where the goal is to find any globally optimal solution.

Droste et al. introduced a model for global optimisation called the *black-box model* [8]. The framework considers search heuristics, called black-box algorithms, that optimise functions over some finite domain  $S$ . The black-box algorithms are limited by the amount of information that is given about the function to be optimised. To obtain the function value of any element in  $S$ , the algorithm needs to make one oracle query. In this framework, lower bounds on the number of queries required for optimisation can be obtained. An advantage of the black-box model is its generality. The model covers any realistic search heuristic. Despite this generality, the lower bounds in the model are in some cases close or equal to the corresponding upper bounds that hold for particular search heuristics. For example, it is shown that the needle-in-the-haystack and trap problems are hard problems, having black-box complexity of  $(2^n + 1)/2$  [8].

However, the black-box model also has some disadvantages. For example, a polynomial black-box complexity of the NP-hard MAX-CLIQUE problem [8] can be shown as follows. A candidate solution is any subset of the vertices. If the vertices in a candidate solution  $S$  form a clique, then the objective value of  $S$  is the cardinality of  $S$ , otherwise the candidate solution is infeasible and the objective value is set to 0. Assume that  $u$  and  $v$  are nodes in the graph. If these nodes are connected, then the candidate solution  $\{u, v\}$  corresponds to a clique of size two. Otherwise, the candidate solution is infeasible, and the objective value is 0. Hence, a black-box algorithm can determine all the edges in a graph of  $n$  nodes by making  $\binom{n}{2}$  queries to the oracle, one query for each pair of nodes. Once the graph instance is known, the optimal solution can be computed offline without further oracle queries. Similar black-box algorithms that uncover the instance can be devised for other NP-hard problems. One cannot expect a realistic search heuristic to solve all instances of an NP-hard problem in polynomial time. Such results exposes two weaknesses: the lower bounds in

the model are often obtained by black-box algorithms that do not resemble any randomised search heuristic. Secondly, the model is too unrestricted with respect to the amount of resources disposable to the algorithm. Black-box algorithms can spend unlimited time in-between queries to do computation.

In order to define a more realistic black-box model, one should consider additional restrictions. Droste et al. suggested to limit the available storage space available to the black-box algorithm [8], but did not prove any lower bounds in the space restricted scenario. Memory-restricted black-box algorithms can be surprisingly efficient. Doerr and Winzen recently proved that even with memory restriction one, the black-box complexity of the ONEMAX function class is at most  $2n$ . Further related work was done by Teytaud and coauthors [10, 30, 31]. Teytaud and Gelly [30] derive lower bounds on the runtime of a broad class of comparison-based randomised search heuristics including evolutionary algorithms and variants of particle swarm optimisation using the decision trees implicitly built by such algorithms. Improvements can be obtained by taking into account the VC-dimension of the level sets induced by the fitness function (Fourniér and Teytaud [10]). Finally, Teytaud, Gelly and Mary [31] investigate a concept called “isotropy” in continuous search spaces, which can be considered analogous to the so-called unbiased mutations studied in this paper. To the best of our knowledge, our model for the first time integrates unbiased mutations and comparison-based algorithms.

The remaining of this paper is organised as follows. Section 3 introduces the new black-box model along with a description of the unbiased variation operators. Section 4 provides the first lower bound in the model for a simple, unimodal problem. Then, in Sect. 5, we consider a function class that contains a plateau. Finally, in Sect. 6, we prove lower bounds that hold for any function with a single, global optimum. The paper is concluded in Sect. 7.

## 2 Preliminaries

The following notation is used. For any integer  $n \geq 1$ ,  $[n]$  denotes the set of integers  $\{1, \dots, n\}$ . The Hamming-distance between two bitstrings  $x, y \in \{0, 1\}^n$  is defined by  $H(x, y) := \sum_{i=1}^n (x_i \oplus y_i)$ , where  $x_i$  denotes the value of the  $i$ -th bit in  $x$ , and  $\oplus$  is the exclusive or-operation. We consider maximisation of pseudo-Boolean functions  $f : \{0, 1\}^n \rightarrow \mathbb{R}$ . In the context of pseudo-Boolean optimisation by randomised search heuristics, the set  $\{0, 1\}^n$  is called the search space, and its elements are called either bitstrings or search points. The LO-value of a bitstring  $x$  is  $\sum_{i=1}^n \prod_{j=1}^i x_j$ , while the LZ-value of  $x$  is  $\sum_{i=1}^n \prod_{j=1}^i (1 - x_j)$ . Informally, the LO-value is the length of the longest prefix of 1-bits in the bitstring, while the LZ-value is the length of the longest prefix of 0-bits in the bitstring. We let  $X \sim p$  signify that  $X$  is a random variable with distribution  $p$ . An event  $\mathcal{E}$  is said to occur *with overwhelming probability* (w.o.p.) with respect to a parameter  $n$  if  $\Pr(\mathcal{E}) = 1 - e^{-\Omega(n)}$ .

We assume that the reader is familiar with the theory of randomised search heuristics (see e.g., [25]). We will frequently refer to an evolutionary algorithm called the  $(\mu + \lambda)$  EA [7, 15, 34]. The parameters  $\mu$  and  $\lambda$  are positive integers. It optimises any pseudo-Boolean function  $f : \{0, 1\}^n \rightarrow \mathbb{R}$  through a sequence of so-called generations. In each generation, the algorithm evaluates the  $f$ -value of  $\lambda$  new search

points. Each new search point to evaluate is chosen uniformly at random among the  $\mu$  search points with highest  $f$ -value from the most recent generation, and flipping each bit independently with probability  $1/n$ . The simple randomised search heuristic Random Local Search (RLS) [23] is defined as the  $(1 + 1)$  EA, except that new search points are obtained by flipping one bit chosen uniformly at random. Note that the  $(\mu + \lambda)$  EA considered here uses mutation only.

### 3 A Refined Black-Box Model

We now present the refined black-box model that is obtained by imposing two additional restrictions in the old black-box model. We start with a preliminary informal description and motivation.

Firstly, we restrict the queries that can be made by the algorithm. The initial query is a bitstring chosen uniformly at random. Every subsequent query must be made for a search point that is obtained by applying a variation operator to one or more of the previously queried search points. We formalise variation operators as conditional probability distributions over the search space. Given  $k$  search points  $x_1, \dots, x_k$ , a variation operator  $p$  produces a search point  $y$  with probability  $p(y \mid x_1, \dots, x_k)$ . Unbiased black-box algorithms are restricted to so-called unbiased variation operators which will be defined in Sect. 3.2. The algorithm is allowed to select a different variation operator in each iteration, as long as the variation operators are statistically independent. More precisely, for any pair of variation operators  $p_1$  and  $p_2$  used by the algorithm, we require that for all bitstrings  $u, v, x_1, \dots, x_k, y_1, \dots, y_k$ ,

$$\begin{aligned} \Pr(X = u \wedge Y = v \mid X \sim p_1(\cdot \mid x_1, \dots, x_k) \wedge Y \sim p_2(\cdot \mid y_1, \dots, y_k)) \\ = \Pr(X = u \mid X \sim p_1(\cdot \mid x_1, \dots, x_k)) \cdot \Pr(Y = v \mid Y \sim p_2(\cdot \mid y_1, \dots, y_k)). \end{aligned}$$

Secondly, we put an additional restriction on the information that is available to the algorithm by preventing the algorithm from observing the bit values of the search points that are queried. Hence, the only information that can be exploited by the algorithm is the sequence of function values obtained when making queries to the black-box, and not the search points themselves. Note that without this restriction, any black-box algorithm could be simulated as follows: For each query  $x$  made by the unrestricted algorithm, the restricted algorithm would solve the sub-problem corresponding of minimising the Hamming distance to search point  $x$ . This sub-problem could be solved analogously to how the  $(1 + 1)$  EA optimises the function  $\text{ONEMAX}(x) := \sum_{i=1}^n x_i$  [7], except that the optimum is chosen to be the desired search point to visit. This simulation would lead to an expected overhead factor of  $O(n \log n)$  function evaluations.

We now turn our considerations into a formal definition of a refined black-box model, as stated in Algorithm 1. Let us first pick up the *unrestricted black-box* model [8] for optimisation of pseudo-Boolean functions  $f : \{0, 1\}^n \rightarrow \mathbb{R}$ . The black-box algorithm  $A$  is given a class of pseudo-Boolean functions  $\mathcal{F}$ . An adversary selects a function  $f$  from this class and presents it to the algorithm as a black-box. At this point, the only information available to the algorithm about the function  $f$  is that it

**Algorithm 1** Unbiased black-box algorithm

- 
- 1:  $t \leftarrow 0$ .
  - 2: Choose  $x(t)$  uniformly at random from  $\{0, 1\}^n$ .
  - 3: **repeat**
  - 4:    $t \leftarrow t + 1$ .
  - 5:   Compute  $f(x(t - 1))$ .
  - 6:    $I(t) \leftarrow (f(x(0)), \dots, f(x(t - 1)))$ .
  - 7:   Depending on  $I(t)$ , choose a probability distribution  $p_s$  on  $\{0, \dots, t - 1\}$ .
  - 8:   Randomly choose an index  $j$  according to  $p_s$ .
  - 9:   Depending on  $I(t)$ , choose an unbiased variation operator  $p_v(\cdot | x(j))$ .
  - 10:   Randomly choose a bitstring  $x(t)$  according to  $p_v$ .
  - 11: **until** termination condition met.
- 

belongs to function class  $\mathcal{F}$ . The black-box algorithm can now start to query an oracle for the function values of any search points. The runtime  $T_{A,f}$  of the algorithm on function  $f$  is the number of function queries on  $f$  until the algorithm queries the function value of an optimal search point for the first time. The runtime  $T_{A,\mathcal{F}}$  on the class of functions is defined as the maximum runtime over the class of functions. In the *unbiased black-box model*, queries of new search points must be made according to Algorithm 1. I.e., the initial search point is chosen uniformly at random by an oracle, and subsequent search points are obtained by asking the oracle to apply a given unbiased variation operator to a previously queried search point.

The *unbiased black-box complexity* of a function class  $\mathcal{F}$  is the minimum worst case runtime  $T_{A,\mathcal{F}}$  among all unbiased black-box algorithms  $A$  satisfying the framework of Algorithm 1. Hence, any upper bound on the worst case of a particular unbiased black-box algorithm, also implies the same upper bound on the unbiased black-box complexity. To prove a lower bound on the unbiased black-box complexity, it is necessary to prove that the lower bound holds for any unbiased black-box algorithm. Note that since unbiased black-box algorithms are a special case of black-box algorithms, all the lower bounds that hold for the unrestricted black-box model by Droste et al. [8] also hold for the unbiased black-box model.

### 3.1 Comparison with the Classical, Unrestricted Model

Introducing unbiasedness to black-box models has important consequences compared to the unrestricted model from [8]. The latter one includes trivial algorithms that only query a specific bitstring, which may be the optimum of a given function (such as the all-ones bitstring for the ONEMAX function). In general, the unrestricted black-box complexity of a class of functions  $\mathcal{F}$  is trivially bounded by  $|\mathcal{F}|$  since the black-box algorithm can query the optima from all functions within the class.

For this reason, Droste et al. [8] generalise the classical example functions to classes of functions where the respective optimum can be any point in the search space. As an example, the class ONEMAX\* consisting of the  $2^n$  functions  $\text{ONEMAX}_a(x) = n - H(x, a)$  for any bitstring  $a \in \{0, 1\}^n$  is considered and the surprisingly low black-box complexity  $\Theta(n/\log n)$  has been proved (see [1] for the upper bound  $O(n/\log n)$ ).

In the unbiased black-box model, algorithms that query a specific search point and then stop are not possible. Here it makes perfect sense to study the unbiased black-box complexity of a particular function. In fact, due to the unbiasedness, the unbiased black-box complexity is the same for every function from the class  $\text{ONEMAX}^*$ .

Note also that the unbiased model is restricted to pseudo-Boolean optimisation and excludes certain search heuristics. More details are given in the following subsections.

### 3.2 Unbiased Variation Operators

To capture the essential characteristics of the variation operators that are used by common randomised search heuristics, we put some restrictions on the probability distribution of the variation operators. Firstly, one can limit the number  $k$  of search points that are used to produce the new search point. The number  $k$  determines the *arity* of the variation operator. Here, we will only consider *unary* variation operators, i.e. when  $k = 1$ . Furthermore, we will impose the following two unbiasedness-conditions on the operators:

- (1)  $\forall x, y, z, p(y \oplus z \mid x_1, \dots, x_k) = p(y \oplus z \mid x_1 \oplus z, \dots, x_k \oplus z)$ ,
- (2)  $\forall x, y, \sigma, p(y \mid x_1, \dots, x_k) = p(\sigma_b(y) \mid \sigma_b(x_1), \dots, \sigma_b(x_k))$ ,

and for any permutation  $\sigma$  over  $[n]$ ,  $\sigma_b$  is an associated permutation over the bitstrings defined as

$$\sigma_b(x_1 x_2 \cdots x_n) := x_{\sigma(1)} x_{\sigma(2)} \cdots x_{\sigma(n)}.$$

Variation operators that satisfy the first condition are called  $\oplus$ -invariant, while variation operators that satisfy the second condition are called  $\sigma$ -invariant. In this paper, an unbiased variation operator is defined as a variation operator that satisfies both conditions, i.e. an  $\oplus$ - $\sigma$ -invariant operator. Note that this is a special case of the framework by Rowe, Vose and Wright [28], who study invariance from a group-theoretical point of view. In a follow-up work to the conference version of this paper [21], Rowe and Vose suggest how unbiasedness-conditions can be generalised to arbitrary search spaces [27].

We claim that the two conditions are natural. Firstly, as we will discuss further in Sect. 3.3, the variation operators used by common randomised search heuristics, including adaptive variation operators, are typically  $\oplus$ - $\sigma$ -invariant. Secondly, the two conditions on the variation operators can also be motivated by practical concerns. In applications, the set of bitstrings typically encode the variable settings of candidate solutions, and is rarely the optimisation domain *per se*. Hence, the encoding from variable setting to bit value, and from variable to bitstring position, can be arbitrary. For example, in an optimisation domain involving a binary temperature parameter, whether the user encodes “high temperature” as 0 or as 1, or decides to encode the temperature variable by the first instead of the last variable in the bitstring, should not influence the behaviour of a search heuristic.

Droste and Wiesmann recommended that all search points that are within the same distance of the originating search point should have the same probability of being produced [9]. This unbiasedness criterion, which we call *Hamming-invariance*, can be formalised as follows:

$$(3) \forall x, y, z, H(x, y) = H(x, z) \implies p(y | x) = p(z | x),$$

where  $H(x, y)$  denotes the Hamming distance between  $x$  and  $y$ . We now show that these criteria are related.

**Proposition 1** *Every unary variation operator that is  $\oplus$ - $\sigma$ -invariant is also Hamming-invariant.*

*Proof* Assume that  $p$  is an  $\oplus$ - $\sigma$ -invariant variation operator. Let  $x, y, u$  and  $v$  be any search points such that  $H(x, y) = H(u, v) =: d$ . Given these assumptions, we will prove that  $p(y | x) = p(v | u)$  holds. There must exist a permutation  $\sigma$  such that  $\sigma(y \oplus x) = v \oplus u$ . We then have

$$\begin{aligned} p(y | x) &= p(y \oplus x | 0^n) = p(\sigma(y \oplus x) | 0^n) \\ &= p(v \oplus u | 0^n) = p(v | u). \end{aligned}$$

By setting  $u = x$ , it is clear that  $p$  is Hamming-invariant. □

When referring to unbiased black-box algorithms in the following, we mean any algorithm that follows the framework of Algorithm 1. In particular, by *unary, unbiased black-box algorithms*, we mean such algorithms that only use statistically independent, unary, unbiased variation operators.

### 3.3 Examples of Unbiased Black-Box Algorithms

The class of unary, unbiased black-box algorithms is general, and contains many well-known randomised search heuristics. For example, simulated annealing [19], random local search (RLS) [23],  $(\mu + \lambda)$  EA [7, 15, 34], and many other population-based EAs that do not use the crossover operator are unbiased black-box algorithms. These algorithms often use the single-bit mutation or bit-wise mutation as variation operators.

The *single-bit mutation* operator is perhaps one of the simplest variation operators, and has been used in simulated annealing, RLS, and other local search heuristics. It modifies a bitstring by flipping one uniformly chosen bit position. This is clearly an unbiased variation operator.

In contrast, evolutionary algorithms typically use the *bit-wise mutation* operator. For any pair of bitstrings  $x, y \in \{0, 1\}^n$ , the operator generates bitstring  $x$  from bitstring  $y$  with probability  $p_{\text{mut}}^{H(x,y)} (1 - p_{\text{mut}})^{n-H(x,y)}$ , i.e., the new bitstring  $x$  is obtained by independently flipping each bit in  $y$  with a probability  $p_{\text{mut}}$ , called the mutation rate. It is straightforward to see that the bit-wise mutation operator is a unary unbiased variation operator for any mutation rate  $p_{\text{mut}}$ . The most common parameter setting is  $p_{\text{mut}} = 1/n$ , but alternative mutation rates have been studied. Some of the *hypermutation* operators used in Artificial Immune Systems (AIS) (see [35] for recent theoretical results), are bit-wise mutation operators with  $p_{\text{mut}}$  chosen as a function of the objective value  $f$  of the parent individual. For example, the mutation rate has been set reciprocal to  $f$ , as in [2], or exponentially decreasing in  $f$ , as in the CLONALG and Opt-aiNet algorithms. As a side note, the related *rank-based mutation* operator

chooses  $p_{\text{mut}}$  as a function of the rank of the parent individual [26]. Another hypermutation operator used in AIS chooses  $p_{\text{mut}}$  proportional to the Hamming-distance to the closest optimal solution. All these variants of the bit-wise mutation operator are unbiased variation operators. However, the latter of these operators cannot be used in the black-box scenario because the Hamming-distance to the set of optimal solutions is generally not available.

Some randomised search heuristics use variation operators that are not unbiased. The *contiguous hypermutation* operator used in AIS randomly selects an interval of consecutive bit positions, and flip each bit in this interval independently with some probability. In the *position-dependant mutation* operator, the bit-flipping probability varies according to the bit position [3]. These two operators are not  $\sigma$ -invariant. In the *asymmetric mutation* operator, the probability of flipping a given bit position depends on the number of bit positions that take the same value [16, 23]. This variation operator is not  $\oplus$ -invariant. A few search heuristics use variation operators that are not statistically independent. For example, the sequences of bit-flips made by *quasirandom mutation* operators are partly deterministic [5]. A memory mechanism is employed so that the partly deterministic sequences share some properties with truly random bit-flipping sequences.

Finally, the restriction to unary operators excludes some randomised search heuristics. In particular, the model does not cover EAs that use crossover. Many of the commonly used diversity mechanisms are excluded. Also, estimation of distribution algorithms [20], ant colony optimisation [6] and particle swarm optimisation [18] are not covered by the model.

#### 4 Simple Unimodal Functions

We consider the unimodal function  $\text{LEADINGONES}(x) = \sum_{i=1}^n \prod_{j=1}^i x_j$  as an initial example. (The function  $\text{ONEMAX}$  is covered by the results in Sect. 6.) The expected runtime of the  $(1 + 1)$  EA on this function is  $\Theta(n^2)$  [7], which seems optimal among commonly analysed EAs. Increasing either the offspring or parent population sizes does not reduce the runtime. For  $(\mu + 1)$  EA, the runtime is  $\Theta(n^2 + \mu n \log n)$  [34], and for  $(1 + \lambda)$  EA, the runtime is  $\Theta(n^2 + \lambda n)$  [15]. In fact, Sudholt showed that a variant of the  $(1 + 1)$  EA is optimal on  $\text{LEADINGONES}$  among a class of algorithms called mutation-based EAs [29].

We now show that the runtime of the  $(1 + 1)$  EA on  $\text{LEADINGONES}$  is asymptotically optimal in the unbiased black-box model. We define the potential of the algorithm at time step  $t$  as the largest number of leading 1- or 0-bits obtained so far. The number of 0-bits must be considered because flipping every bit in a bitstring with  $i$  leading 0-bits will produce a bitstring with  $i$  leading 1-bits. The increase of the potential will be studied using drift analysis [12, 13]. We use the following variant of the polynomial drift theorem due to [14].

**Theorem 1** ([14]) *Let  $\{X^{(t)}\}_{t \geq 0}$  be a sequence of random variables with bounded support and let  $T$  be the stopping time defined by  $T := \min\{t \mid X^{(1)} + \dots + X^{(t)} \geq g\}$  for a given  $g > 0$ . If  $\mathbf{E}[T]$  exists and  $\mathbf{E}[X^{(i)} \mid T \geq i] \leq u$  for  $i \in \mathbb{N}$ , then  $\mathbf{E}[T] \geq g/u$ .*



To lower bound the drift, it is helpful to proceed as in the analysis of  $(1 + 1)$  EA on LEADINGONES [7], i.e., to first prove that the substring after the first 0-bit in a given time step is uniformly distributed. We will prove a more general statement that will also be used in Sect. 5. For notational convenience, subsets of  $[n]$  and bitstrings of length  $n$  will be used interchangeably, i.e., the bitstring  $x \in \{0, 1\}^n$  is associated with the subset  $\{i \in [n] \mid x_i = 1\}$ .

**Lemma 1** *For any  $t \geq 0$ , let  $X(t) = \{x(0), x(1), \dots, x(t)\}$  be the search points visited by any unary, unbiased black-box algorithm until iteration  $t$  when optimising function  $f$ . If there exists a subset of indices  $z \subseteq [n]$  such that*

$$\forall y \subseteq z, \forall x \in X(t) \quad f(x \oplus y) = f(x),$$

*then the bits  $\{x_i(t) \mid i \in z\}$  are independent and uniformly distributed.*

*Proof* We first prove the claim that for all  $t \geq 0$ ,  $b \in \{0, 1\}$ , and  $i \in z$  it holds that  $\Pr(x_i(t) = b) = 1/2$ . The proof is by induction over the time step  $t$ . The initial search point is sampled uniformly at random, so the claim holds for  $t = 0$ . Assume that the claim holds for  $t = t_0 \geq 0$ . The function value profile  $I(t_0)$  does not depend on any of the bits in position  $i$  in the previously visited search points. The choice of search point  $x(j)$  the algorithm makes is therefore independent of these bits. By the induction hypothesis, search point  $x(j)$  has a bit value of  $b$  in position  $i$  with probability  $p := 1/2$ . Letting  $r$  be the probability that the variation operator flips bit position  $i$ , we have  $\Pr(x_i(t_0 + 1) = b) = p(1 - r) + (1 - p)r = 1/2$ . By induction, the claim now holds for all  $t$ .

We now prove the lemma by induction over the time  $t$ . The lemma holds for time step  $t = 0$ . Assume that the lemma holds for  $t = t_0 \geq 0$ . Let  $U \in \{0, 1\}^n$  be a random vector where the elements  $U_i$ ,  $1 \leq i \leq n$ , take the value  $U_i = 1$  if bit position  $i$  flipped in step  $t_0$ , and  $U_i = 0$  otherwise. Then for all bitstrings  $y \in \{0, 1\}^n$ ,

$$\Pr\left(\bigwedge_{i \in z} x_i(t_0 + 1) = y_i\right) = \sum_{u \in \{0, 1\}^n} \Pr(U = u) \Pr\left(\bigwedge_{i \in z} x_i(t_0) = y_i \oplus u_i\right),$$

which by the induction hypothesis (more precisely, the statement about independence) equals

$$\sum_{u \in \{0, 1\}^n} \Pr(U = u) \prod_{i \in z} \Pr(x_i(t_0) = y_i \oplus u_i) = 2^{-|z|} = \prod_{i \in z} \Pr(x_i(t_0 + 1) = y_i),$$

where the last equality follows from the claim above. This proves the independence at time  $t_0 + 1$ , and, therefore, the induction step.  $\square$

**Theorem 2** *The expected runtime of any unary, unbiased black-box algorithm on LEADINGONES is  $\Omega(n^2)$ .*

*Proof* Recall that the potential of the algorithm at time step  $t$  is defined as  $k := \max_{0 \leq j \leq t} \{\text{LO}(x(j)), \text{LZ}(x(j))\}$ , where LO and LZ are defined in Sect. 2.

We first prove the claim that w.o.p., the potential of the algorithm will at some time be in the interval between  $n/2$  and  $3n/4$ , before the optimum has been found. Since the initial search point is uniform, the probability of initialising with at least  $k$  leading ones or least  $k$  leading zeros is at most  $2 \cdot 2^{-k} = 2^{-k+1}$ . Hence, with probability  $1 - 2^{-n/2+1}$ , the initial search point will have potential less than  $n/2$ . Let integer  $i$  be the number of 0-bits in the interval from  $n/2$  and  $3n/4$  in the bitstring that the algorithm selects next. By Lemma 1 with the index set  $z := [n/2 + 1, n]$ , and a Chernoff bound, there is a constant  $\delta > 0$  such that w.o.p., this integer satisfies  $(1 - \delta)n/8 < i < (1 + \delta)n/8$ . In order to increase the LO-value from less than  $n/2$  to at least  $3n/4$ , it is necessary to flip every 0-bit in the interval from  $n/2$  to  $3n/4$  and no other bits. We optimistically assume that exactly  $i$  bits are flipped in this interval. However due to the unbiasedness condition, every choice of  $i$  among  $n/4$  bits to flip is equally likely. The probability that only the 0-bits are flipped is therefore at most  $\binom{n/4}{i}^{-1} \leq (4i/n)^i \leq ((1 + \delta)/2)^{(1-\delta)n/8}$ . An analogous argumentation applies if  $i$  denotes the number of 1-bits in the interval, such that also the LZ-value is bounded by  $3n/4$  with overwhelming probability. The claim therefore holds.

We will apply drift analysis according to the potential  $k$  of the algorithm, only counting the steps starting from a potential in the interval  $n/2 \leq k < 3n/4$ . In order to find the optimum, the potential must be increased by at least  $n/4$ . We now inspect the probability of increasing the best-so-far LO-value from  $k$  to some larger value, noting that an analogous argument holds for the LZ-value. Assume that the selected search point has  $r \geq 1$  0-bits in the first  $k + 1$  bit positions. In order to increase the LO-value, it is necessary to flip all these 0-bits, and none of the 1-bits within this interval. The unbiased mutation operator has to flip a uniform subset of  $\{1, \dots, k + 1\}$  of some size  $s$ . If  $s < r$  then the best-so-far LO-value cannot be increased. Otherwise, we model the process of selecting the subset as consecutively and without replacement drawing from an urn containing a ball for each of the bit positions  $1, \dots, k + 1$ . If a position is 1, the corresponding ball is white, otherwise it is red. In order to flip only the 0-bits, it is necessary to draw in the first  $r$  trials all the  $r$  red balls. The probability of this event is less than

$$\frac{r}{k+1} \cdot \frac{r-1}{k} \cdots \frac{2}{k-r+2} \cdot \frac{1}{k-r+1} \leq \frac{1}{k+1}.$$

By a union bound, the probability of increasing the LO-value or the LZ-value, hence the probability of increasing the potential, is at most  $2/(k + 1)$ .

Considering an increase of the LO-value, it is important to note that the search point produced in iteration  $t$  may contain one or more 1-bits after position  $k + 1$ . The consecutive 1-bits in position  $k + 2$  and on-wards in the search point produced in iteration  $t$  are called *free-riders* [7]. Analogously, there may be free-riders when increasing the LZ-value. Let random variable  $Y^{(t)}$  denote the number of free-riders (either ones or zeros, depending on the value of  $k + 1$ ) in iteration  $t$ . Since the probability of increasing the potential has been bounded by  $2/(k + 1)$ , the drift of the potential in each step is bounded from above by  $\Delta(t) \leq (2/k + 1)(1 + \mathbf{E}[Y^{(t)}])$ . Applying Lemma 1 with the index set  $z := [k + 2, n]$  gives  $\mathbf{E}[Y^{(t)}] \leq \sum_{i=1}^{\infty} 2^{-i} \leq 1$  no matter if the free-rider are ones or zeros. Altogether,  $\Delta(t) \leq 4/(k + 1)$ . Let  $T$

denote the time until the potential has increased to  $n$ . The polynomial drift theorem (Theorem 1) now implies that  $\mathbf{E}[T] \geq (n/4)/\Delta(t) = (k + 1)n/16 = \Omega(n^2)$ .  $\square$

Note that the complexity of LEADINGONES in the unrestricted black-box model (where, as justified in Sect. 3.1, a generalised class of functions is analysed) is bounded above by  $n/2 - o(n)$  [8]. This illustrates that the complexity of a function class can be significantly higher in the unbiased black-box model than in the unrestricted black-box model.

### 5 Enforcing Expected Runtimes

We are interested in problem classes where the unbiased black-box complexity depends on some parameter of the problem. More specifically, is there a class of functions  $\mathcal{F} = \{f_i \mid i \in \mathbb{Z}\}$ , such that the unbiased black-box complexity of the functions  $f_m$  in the class  $\mathcal{F}$  increases with the problem parameter  $m$ , e.g. as  $\Theta(2^m)$ ?

In the case of the  $(1 + 1)$  EA, it is known that the runtime can depend on the size of so-called plateaus. A plateau is a contiguous region of the search space where the fitness function is constant. Plateaus are common in combinatorial optimisation problems. For example, on certain inputs to the maximum matching problem, the  $(1 + 1)$  EA has to traverse a plateau of size  $n$  to make an improvement [11]. This is accounted for by an  $\Omega(n^2)$  factor in the expected runtime. Another example is the NEEDLE function where the whole search space except for the optimum is a plateau. Here the expected runtime of the  $(1 + 1)$  EA is  $\Omega(2^n)$ .

We are aiming at generalising this result to black-box algorithms with unary unbiased variation. Moreover, we want to enforce expected running times that are of order  $2^m$  for a parameter  $m$ . To achieve this, we pick up the general idea of the NEEDLE function in [7], but modify the “plateau”. For any  $m$ , where  $1 \leq m \leq n$ , define

$$\text{OM-NEEDLE}_m(x) := \sum_{i=1}^{n-m} x_i + \prod_{i=1}^n x_i.$$

Hence, the plateau corresponds to the last  $m$  bit positions which do not influence the function value, except at the optimal search point  $1^n$ . For  $m = 1$ , we obtain the easy ONEMAX function, and for  $m = n$ , we obtain the hard NEEDLE function. The following result shows that the hardness of OM-NEEDLE $_m$  depends on the problem parameter  $m$ .

**Theorem 3** *For  $0 \leq m \leq n$ , the expected runtime of any unary, unbiased black-box algorithm on OM-NEEDLE $_m$  is at least  $2^{m-2}$ . Furthermore, the probability that the optimum is found within  $2^{m(1-\varepsilon)}$  iterations for any  $\varepsilon, 0 < \varepsilon < 1$ , is no more than  $2^{-\varepsilon m}$ .*

*Proof* The time  $T$  to find the optimum is bounded from below by the time  $T'$  until all the last  $m$  bits are 1-bits. The time is analysed as if the algorithm was presented with the function  $f(x) = \sum_{i=1}^{n-m} x_i$  instead of OM-NEEDLE $_m$ . This function differs

from OM-NEEDLE<sub>m</sub> only on the optimal search point 1<sup>n</sup>. The distribution of T' will therefore be the same for f and OM-NEEDLE<sub>m</sub>.

Lemma 1 now applies for the function f and the set of indices z = [n - m + 1, n]. The probability that the search point visited in any given iteration has 1<sup>m</sup> as suffix is 2<sup>-m</sup>. By union bound, the probability that this suffix is obtained before iteration t ≥ 0 is no more than t · 2<sup>-m</sup>. In particular, the probability that the runtime is shorter than 2<sup>m-1</sup> is less than 1/2, so the expected runtime is at least 2<sup>m-2</sup>. Furthermore, the probability that the optimum has been found within 2<sup>m(1-ε)</sup> iterations is no more than 2<sup>-εm</sup>. □

We supplement an upper bound.

**Theorem 4** *There exists a unary, unbiased black-box algorithm whose expected runtime on OM-NEEDLE<sub>m</sub> is O((n/m) · 2<sup>m</sup> + n log n).*

*Proof* We show that the well-known algorithm random local search (RLS) [23] has the stated expected runtime. For the purpose of the analysis, we assume that the algorithm optimises the function f(x) = ∑<sub>i=1</sub><sup>n-m</sup> x<sub>i</sub>, but still consider the first point of time such that 1<sup>n</sup> is sampled. Note that f only differs from OM-NEEDLE<sub>m</sub> on the optimal search point 1<sup>n</sup>. Hence this change in the analysis will not have any impact on the runtime distribution of the algorithm.

During a *first phase*, the algorithm will be able to increase the function value by gaining 1-bits in the first n - m positions. By a coupon collector argument [22], the expected time of this phase is O(n log n).

In a *second phase*, the function value of the current search point is n - m, and the algorithm will only accept a new search point if it was obtained by flipping one of the m bits in the suffix. We call such steps *good*. The analysis follows the ideas in Sect. 3 in [33]. Since we consider the function f, all the search points on the form 1<sup>n-m</sup>x, x ∈ {0, 1}<sup>m</sup>, have function value n - m. Hence, the Markov chain corresponding to the suffix is ergodic with the uniform distribution as stationary distribution. This means that if we consider the Markov process X<sup>(t)</sup> ∈ [0, m], where X<sup>(t)</sup> denotes the number of 1-bits in the suffix, then each state i ∈ [0, m] has stationary distribution π(i) =  $\binom{m}{i} \cdot 2^{-m}$ . For i, j ∈ [0, m], let T<sub>ij</sub> be the first hitting time from state i to state j, and p(i, j) the transition probability from state i to state j. By the fundamental theorem of ergodic Markov chains [22], E[T<sub>ii</sub>] = 1/π(i). Hence, by the law of total probability,

$$p(j + 1, j) \cdot \mathbf{E}[T_{j,j+1}] \leq \mathbf{E}[T_{j+1,j+1}] = \frac{1}{\pi(j + 1)} = \frac{2^m}{\binom{m}{j+1}}.$$

The optimisation time T is bounded from above by the sum of the hitting times T<sub>j,j+1</sub> for 0 ≤ j ≤ n - 1. Hence,

$$\mathbf{E}[T] \leq (n/m) \sum_{j=0}^{m-1} \mathbf{E}[T_{j,j+1}] \leq (n/m) \sum_{j=0}^{m-1} \frac{2^m}{p(j + 1, j) \binom{m}{j+1}},$$

where the  $n/m$  factor accounts for the waiting time for a good step. By noting that  $p(j + 1, j) = (j + 1)/m$ , the denominators can be simplified as

$$\binom{m}{j+1} \cdot \frac{j+1}{m} = \binom{m-1}{j}.$$

Since the binomial coefficient takes its smallest value for  $j = 0$  and  $j = m - 1$  and its second-smallest one for  $j = 1$  (and  $j = m - 2$ ), we have

$$\sum_{j=0}^{m-1} \frac{1}{\binom{m-1}{j}} \leq \frac{1}{\binom{m-1}{0}} + \frac{1}{\binom{m-1}{m-1}} + \sum_{j=1}^{m-2} \frac{1}{\binom{m-1}{1}} \leq 2 + \frac{m-2}{m-1} \leq 3.$$

Thus, the expected duration of the second phase is  $O((n/m) \cdot 2^m)$ .

The theorem now follows by adding the expected times of the two phases. □

### 6 General Functions

In the previous sections, we provided bounds on particular pseudo-Boolean functions that are commonly considered in the runtime analysis of randomised search heuristics. In this section, we focus on finding lower bounds that hold for any function. Such bounds are only interesting when we consider functions that correspond to realistic optimisation problems, as trivial functions like constant functions can be optimised with a single function evaluation. We therefore focus on functions that have a unique global optimum.

It is of interest to compare the lower bounds in the black-box models with those bounds that have been obtained for specific EAs. Wegener proved a lower bound of  $\Omega(n \log n)$  for the  $(1 + 1)$  EA on any function with a unique optimum [32]. This bound is significantly larger than the  $\Omega(n/\log n)$  bound that holds for the generalisation ONEMAX\* of the ONEMAX problem mentioned in Sect. 3.1. Given this discrepancy, one can ask whether there is room to design better EAs which overcome the  $n \log n$  barrier, or whether the black-box bound is too loose. Jansen et al. provided evidence that there is little room for improvement by showing that any EA that uses uniform initialisation, selection and bit-wise mutation with probability  $1/n$  needs  $\Omega(n \log n)$  function evaluations to optimise functions with a unique optimum [15].

In the following, we will generalise this result further, showing that the  $n \log n$ -barrier for functions of a unique optimum even holds for the wider class of unary, unbiased black-box algorithms. The idea behind the proof is to show that the probability of making an improving step reduces as the algorithm approaches the optimum. To implement this idea, we will apply Theorem 5. This is a lower-bound analogue to a technique which is called expected multiplicative weight decrease in the evolutionary computation literature [24]. Theorem 5 will be proved using the polynomial drift theorem (Theorem 1), and the following simple lemma.

**Lemma 2** *Let  $X$  be any random variable, and  $k$  any real number. If it holds that  $\Pr(X < k) > 0$ , then  $\mathbf{E}[X] \geq \mathbf{E}[X \mid X < k]$ .*

*Proof* Define  $p := \Pr(X < k)$  and  $\mu_k := \mathbf{E}[X \mid X < k]$ . The lemma clearly holds when  $p = 1$ , so we consider the case where  $0 < p < 1$ . If  $\mathbf{E}[X]$  is positive infinite, then the theorem clearly holds because  $\mu_k < k$ . If  $\mathbf{E}[X]$  is negative infinite, then so is  $\mu_k$  by the law of total probability. Finally, for finite  $\mathbf{E}[X]$ , we have by the law of total probability

$$\begin{aligned} \mathbf{E}[X] &= (1 - p) \cdot \mathbf{E}[X \mid X \geq k] + p \cdot \mu_k \\ &\geq (1 - p) \cdot k + p \cdot \mu_k \\ &> (1 - p) \cdot \mu_k + p \cdot \mu_k \\ &= \mathbf{E}[X \mid X < k]. \end{aligned} \quad \square$$

**Theorem 5 (Multiplicative Drift, Lower Bound)** *Let  $S \subseteq \mathbb{R}$  be a finite set of positive numbers with minimum 1. Let  $\{X^{(t)}\}_{t \geq 0}$  be a sequence of random variables over  $S$ , satisfying  $X^{(t+1)} \leq X^{(t)}$  for any  $t \geq 0$ , and let  $s_{\min} \geq 1$ . Let  $T$  be the random first point in time  $t \geq 0$  for which  $X^{(t)} \leq s_{\min}$ . If there exist positive reals  $\beta < 1$  and  $\delta \leq 1$  such that for all  $s > s_{\min}$  and all  $t \geq 0$  with  $\Pr(X^{(t)} = s) > 0$  it holds that*

1.  $\mathbf{E}[X^{(t)} - X^{(t+1)} \mid X^{(t)} = s] \leq \delta s$ ,
2.  $\Pr(X^{(t)} - X^{(t+1)} \geq \beta s \mid X^{(t)} = s) \leq \beta \delta / \ln s$ ,

then for all  $s_0 \in S$  with  $\Pr(X^{(0)} = s_0) > 0$ ,

$$\mathbf{E}[T \mid X^{(0)} = s_0] \geq \frac{\ln(s_0) - \ln(s_{\min})}{\delta} \cdot \frac{1 - \beta}{1 + \beta}.$$

*Proof* The proof generalises the proof of Theorem 1 in [5]. The random variable  $T$  is non-negative. Hence, if the expectation of  $T$  does not exist, then it is positive infinite and the theorem holds. We condition on the event  $T > t$ , but we omit stating this event in the expectations for notational convenience. We define the stochastic process  $Y^{(t)} := \ln(X^{(t)})$  (note that  $X^{(t)} \geq 1$ ), and apply Theorem 1 with respect to the random variables

$$\Delta_{t+1}(s) := (Y^{(t)} - Y^{(t+1)} \mid X^{(t)} = s) = \left( \ln\left(\frac{s}{X^{(t+1)}}\right) \mid X^{(t)} = s \right).$$

We consider the time until  $X^{(t)} \leq s_{\min}$  if  $X^{(0)} = s_0$ , and therefore set the first parameter in Theorem 1 to  $g := \ln(s_0/s_{\min})$ . To obtain the desired bound, we set the second parameter of Theorem 1 to  $u := \delta(\beta + 1)/(1 - \beta)$ .

By the law of total probability, the expectation of  $\Delta_{t+1}(s)$  can be expressed as

$$\begin{aligned} &\Pr(s - X^{(t+1)} \geq \beta s) \cdot \mathbf{E}[\Delta_{t+1}(s) \mid s - X^{(t+1)} \geq \beta s] \\ &+ \Pr(s - X^{(t+1)} < \beta s) \cdot \mathbf{E}[\Delta_{t+1}(s) \mid s - X^{(t+1)} < \beta s]. \end{aligned} \quad (1)$$

By applying the second condition from the theorem, the first term in (1) can be bounded from above by  $\frac{\beta \delta}{\ln s} \cdot \ln s = \beta \delta$ . The logarithmic function is concave. Hence,

by Jensen’s inequality, the second term in (1) is at most

$$\begin{aligned} & \ln\left(\mathbf{E}\left[\frac{s}{X^{(t+1)}} \mid s - X^{(t+1)} < \beta s \wedge X^{(t)} = s\right]\right) \\ &= \ln\left(1 + \mathbf{E}\left[\frac{s - X^{(t+1)}}{X^{(t+1)}} \mid s - X^{(t+1)} < \beta s \wedge X^{(t)} = s\right]\right). \end{aligned}$$

By using the inequality  $\ln(1 + x) \leq x$  and the conditions  $X^{(t+1)} \geq (1 - \beta)s$  and  $X^{(t+1)} \leq X^{(t)}$ , this simplifies to

$$\begin{aligned} & \mathbf{E}\left[\frac{s - X^{(t+1)}}{X^{(t+1)}} \mid s - X^{(t+1)} < \beta s \wedge X^{(t)} = s\right] \\ & < \mathbf{E}\left[\frac{s - X^{(t+1)}}{(1 - \beta)s} \mid s - X^{(t+1)} < \beta s \wedge X^{(t)} = s\right]. \end{aligned}$$

By Lemma 2 and the first condition from the theorem, it follows that the second term in (1) is at most

$$\mathbf{E}\left[\frac{s - X^{(t+1)}}{(1 - \beta)s} \mid X^{(t)} = s\right] \leq \frac{\delta}{1 - \beta}.$$

Altogether, we obtain  $\mathbf{E}[\Delta_{t+1}(s)] \leq (\beta + 1/(1 - \beta))\delta \leq ((\beta + 1)/(1 - \beta))\delta$ . From Theorem 1, it now follows that

$$\mathbf{E}[T \mid X^{(0)} = s_0] \geq \frac{1}{\delta} \cdot \frac{1 - \beta}{1 + \beta} \cdot \ln\left(\frac{s_0}{s_{\min}}\right). \quad \square$$

We have seen in the proof of Theorem 2 that it is helpful to model the application of an unbiased variation operator as a classical urn experiment. We take a closer look at this model now. Assume that the black-box algorithm chooses a search point  $x$  that has  $m$  0-bits and that the variation operator creates a new search point  $x'$  by flipping  $r$  bits in  $x$ . This corresponds to drawing  $r$  balls without replacement from an urn containing  $m$  red balls and  $n - m$  white balls. The number of red balls  $Z$  in the sample, i.e., the number of flipped 0-bits, is a *hypergeometrically* distributed random variable with expectation  $rm/n$ . If the optimal search point is  $1^n$ , then the change in Hamming-distance to the optimum can be expressed as  $H(x, 1^n) - H(x', 1^n) = Z - (r - Z) = 2Z - r$ . So the shortest distance from the sampled points to the optimum can only decrease if  $Z > r/2$ . Hence, in order to apply Theorem 5, it will be helpful to have an estimate of the expectation of a hypergeometric random variable, conditional on the event that this variable takes at least a certain value.

**Lemma 3** *Let  $Z$  be a hypergeometrically distributed random variable with parameters  $n$  (number of balls),  $r$  (number of samples) and  $m$  (number of red balls), then for all  $k, 0 \leq k \leq r$ ,  $\mathbf{E}[Z \mid Z \geq k] \leq k + (r - k)(m - k)/(n - k)$ .*

*Proof* Let random variable  $S$  be the number of samples needed to obtain  $k$  red balls assuming that  $Z \geq k$ , and let  $Y$  be the number of additional red balls obtained in the

remaining  $r - S$  samples. Then  $Y$  is a hypergeometrically distributed random variable with parameters  $n - S$ ,  $r - S$ , and  $m - k$ . By the law of total probability, it holds that

$$\mathbf{E}[Z \mid Z \geq k] = k + \sum_{s=k}^r \Pr(S = s) \mathbf{E}[Y \mid S = s] \tag{2}$$

$$= k + \sum_{s=k}^r \Pr(S = s) \frac{(r - s)(m - k)}{n - s}. \tag{3}$$

The derivative of the function  $f(s) := (r - s)/(n - s)$  is

$$f'(s) = \frac{1}{n - s} \left( \frac{r - s}{n - s} - 1 \right).$$

Hence,  $f'(s) \leq 0$  whenever  $0 \leq r \leq n$  and  $s \geq 0$ . This implies that the function  $f$  is non-increasing in  $s$ , and we can simplify the right hand side of (3) by fixing  $s = k$ ,

$$\mathbf{E}[Z \mid Z \geq k] \leq k + \frac{(r - k)(m - k)}{n - k} \sum_{s=k}^r \Pr(S = s),$$

from which the lemma follows. □

We also need upper bounds on the tail of the hypergeometric distribution. The following result due to Chvátal [4] is an analogue to the Chernoff bounds for the binomial distribution.

**Lemma 4** ([4]) *If  $X$  is a hypergeometrically distributed random variable with parameters  $n$  (number of balls),  $m$  (number of red balls), and  $r$  (number of samples), then  $\Pr(X \geq \mathbf{E}[X] + r\delta) \leq \exp(-2\delta^2 r)$ , where  $\mathbf{E}[X] = \frac{rm}{n}$ .*

We now state the main result of this section.

**Theorem 6** *The expected runtime of any unary, unbiased black-box algorithm on any pseudo-Boolean function with a single global optimum is  $\Omega(n \log n)$ .*

*Proof* Without loss of generality, assume that the search point  $1^n$  is the single global optimum. The potential  $P^{(t)}$  of the algorithm in a given iteration  $t$  is defined as the shortest Hamming-distance from any previously sampled search point to this optimum. Flipping every bit in a search point with  $s$  0-bits clearly creates a search point with  $n - s$  0-bits. It therefore simplifies the analysis to assume that every time the algorithm applies a variation operator to a bitstring, it also applies the same variation operator to the complementary bitstring. The potential is now bounded to the interval  $[0, n/2]$ , and the optimum has not been found before the potential has reached the value 0.

We call a run *typical* if the potential does not decrease below  $n/10$  without first reaching the interval  $[n/10, n/5)$ . We claim that a run is typical with overwhelmingly



high probability, and that the expected runtime of a typical run is  $\Omega(n \log n)$ . If both claims are correct, then the theorem follows by the law of total probability.

We bound the runtime of typical runs using Theorem 5 with respect to the process  $\{P^{(t)}\}_{t \geq 0}$ . Clearly, the process is monotonically decreasing and has bounded support. We only account for the time interval from some point in time  $t_0$  when the potential is  $P^{(t_0)} = s_0 \in (n/10, n/5]$ , to the first point in time when the potential is below  $s_{\min} := \lceil (50/\beta)(\ln(2/\beta\delta) + \ln \ln n) \rceil$ , where  $\beta \in (0, 1)$  is an arbitrary constant. If the two conditions of Theorem 5 hold for the parameters  $\delta := 200/n$ ,  $\beta$ ,  $s_{\min}$ , and  $s_0$ , then the expected runtime of a typical run is  $\Omega(n \log n)$ .

We verify the *second condition* of Theorem 5 first. Assume that  $P^{(t)} = s$ , where  $s \in (s_{\min}, n/5)$  because the run is typical. Consider first the case where the number of 0-bits in the selected search point is  $m \in [s, n/2]$ . Let  $r \geq 1$  be the number of bits that were flipped by the variation operator. The number of 0-bits that are flipped by the variation operator in iteration  $t$  is a hypergeometrically distributed random variable  $Z^{(t)}$  with parameters  $n$  (number of balls),  $m$  (number of red balls), and  $r$  (number of samples). The reduction in potential can now be expressed as  $\max\{Z^{(t)} - (r - Z^{(t)}) - (m - s), 0\}$ . The potential can therefore only reduce when

$$Z^{(t)} > \frac{r + m - s}{2} =: k. \tag{4}$$

We claim that  $k \geq \mathbf{E}[Z^{(t)}] + r/10$ , where  $\mathbf{E}[Z^{(t)}] = rm/n$ . The claim is true when  $m \in [s, 4n/10]$ , because  $r/2 \geq rm/n + r/10$ , and  $m - s \geq 0$ . The claim is also true when  $m \in (4n/10, n/2]$ , because  $r/2 \geq rm/n$ , and  $m - s \geq n/5 \geq r/5$ . So, by Lemma 4, the probability  $p(s, r, m)$  of reducing the potential when flipping  $r$  bits in a search point with  $m$  0-bits satisfies

$$p(s, r, m) \leq \Pr\left(Z^{(t)} \geq \mathbf{E}[Z^{(t)}] + \frac{r}{10}\right) \leq \exp\left(-\frac{r}{50}\right). \tag{5}$$

We now claim that (5) also holds in the case when  $m \in (n/2, n - s]$  because of the following symmetry argument. The process of creating a new search point by flipping  $r$  bits can be divided into two steps. An intermediate search point is first created by flipping all bits. Then, the final search point is obtained by flipping  $n - r$  bits. The intermediate search point will have  $m' := n - m \in [s, n/2]$  0-bits with probability 1. Inequality (5) therefore applies to the intermediate search point, and the claim holds.

In order to reduce the potential by at least  $s\beta$  0-bits, it is necessary to flip  $r \geq \beta s \geq \beta s_{\min}$  bit positions. The probability that this occurs from a bitstring with  $m$  0-bits, or from the complementary bitstring with  $n - m$  0-bits is by the union bound no more than

$$p(s, r, m) + p(s, r, n - m) \leq 2 \exp\left(-\frac{\beta s_{\min}}{50}\right) \leq \frac{\beta\delta}{\ln n} \leq \frac{\beta\delta}{\ln s},$$

and the second condition of Theorem 5 holds.

We now verify the *first condition*. It is necessary to flip at least  $r \geq m - s$  bits to obtain a search point with less than  $s$  0-bits from a search point with  $m$  0-bits. Equation (4) implies the upper bound  $m - k \leq s$ , which when used together with

Lemma 3 means that the expected reduction in potential conditional on the event  $Z^{(t)} > k$  can be bounded from above as

$$\mathbf{E}[2Z^{(t)} - 2k \mid Z^{(t)} > k] \leq \frac{2(r-k)(m-k)}{n-k} \leq \frac{2(r-k)s}{n-k} \leq \frac{2rs}{n}.$$

The last inequality is valid, because the function  $f(k) = (r-k)/(n-k)$  is decreasing in  $k$ , as shown in the proof of Lemma 3. Hence, by the law of total probability, the unconditional expected decrease in potential by the selected bitstring, and its complement, is no more than

$$(p(s, r, m) + p(s, r, n - m)) \cdot \frac{2rs}{n} \leq \frac{4rs}{ne^{r/50}} \leq \frac{4rs}{n(1+r/50)} \leq \frac{200s}{n} = \delta s. \quad (6)$$

The first condition of Theorem 5 is therefore satisfied.

Finally, we prove the claim that w.o.p., the potential will at some point in time be in the interval  $[n/10, n/5]$ . By a Chernoff bound, the initial potential is w.o.p. in the interval  $(n/5, n/2)$  because the initial search point and its complement are chosen uniformly at random.

In order to reduce the potential from the interval  $[n/2, n/5)$  to the interval  $[0, n/10)$  in one iteration, it is clearly necessary to flip  $r \geq n/10$  bits. Furthermore, following the derivation of (4), it is necessary that the number  $Z^{(t)}$  of 0-bits flipped is strictly larger than  $k' := (r + n/10)/2$ . If the selected search point has  $m \in [n/5, n/2]$  0-bits, then  $k' \geq rm/n + n/20 \geq \mathbf{E}[Z^{(t)}] + r/20$ . By Lemma 4, it then holds that  $\Pr(Z^{(t)} > k') = e^{-\Omega(r)}$ , and this bound can be extended to all  $m \in [n/5, 4n/5]$  due to the symmetry argument above. By a union bound, the probability that the potential is reduced below  $n/10$  by flipping  $r \geq n/10$  bits in the selected search point or the complementary search point is less than  $2\Pr(Z^{(t)} > k') = e^{-\Omega(n)}$ . The claim about the probability of typical runs therefore holds, and the theorem follows.  $\square$

## 7 Conclusions

This paper takes a step forward in building a unified theory of randomised search heuristics. We have defined a new black-box model that captures essential aspects of randomised search heuristics. The new model covers many of the common search heuristics, including simulated annealing and EAs commonly considered in theoretical studies. We have proved upper and lower bounds on the runtime of several commonly considered pseudo-Boolean functions. For some functions, the lower bounds coincide with the upper bounds for the  $(1+1)$  EA, implying that this simple EA is asymptotically optimal on the function class. It is shown that any search heuristic in the model needs  $\Omega(n \log n)$  function evaluations to optimise functions with a unique optimum. Also, it is shown that a function with a plateau can pose a difficulty for any black-box search heuristic in the model.

This work can be extended in several ways. Firstly, it is interesting to consider more problem classes than those considered here. Secondly, the analysis should be extended to variation operators with greater arity than one. Finally, alternative black-box models could be defined that cover ant colony optimisation, particle swarm optimisation, and estimation of distribution algorithms.

## References

- Anil, G., Wiegand, R.P.: Black-box search by elimination of fitness functions. In: Proceedings of the 10th International Workshop on Foundations of Genetic Algorithms (FOGA'09), pp. 67–78. ACM Press, New York (2009)
- Böttcher, S., Doerr, B., Neumann, F.: Optimal fixed and adaptive mutation rates for the leadingones problem. In: Proceedings of the 11th international conference on Parallel Problem Solving from Nature (PPSN'10), pp. 1–10. Springer, Berlin (2010)
- Cathabard, S., Lehre, P.K., Yao, X.: Non-uniform mutation rates for problems with unknown solution lengths. In: Proceedings of the 11th International Workshop on Foundations of Genetic Algorithms (FOGA'11), pp. 173–180. ACM, New York (2011)
- Chvátal, V.: The tail of the hypergeometric distribution. *Discrete Math.* **25**(3), 285–287 (1979)
- Doerr, B., Fouz, M., Witt, C.: Quasirandom evolutionary algorithms. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation (GECCO'10), pp. 1457–1464. ACM, New York (2010)
- Dorigo, M., Stützle, T.: *Ant Colony Optimization*. MIT Press, Cambridge (2004)
- Droste, S., Jansen, T., Wegener, I.: On the analysis of the (1 + 1) Evolutionary Algorithm. *Theor. Comput. Sci.* **276**, 51–81 (2002)
- Droste, S., Jansen, T., Wegener, I.: Upper and lower bounds for randomized search heuristics in black-box optimization. *Theory Comput. Syst.* **39**(4), 525–544 (2006)
- Droste, S., Wiesmann, D.: Metric based evolutionary algorithms. In: Proceedings of Genetic Programming, European Conference. LNCS, vol. 1802, pp. 29–43. Springer, Berlin (2000)
- Fournier, H., Teytaud, O.: Lower bounds for comparison based evolution strategies using VC-dimension and sign patterns. *Algorithmica* **59**, 387–408 (2011)
- Giel, O., Wegener, I.: Evolutionary algorithms and the maximum matching problem. In: Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science (STACS'03), pp. 415–426 (2003)
- Hajek, B.: Hitting-time and occupation-time bounds implied by drift analysis with applications. *Adv. Appl. Probab.* **13**(3), 502–525 (1982)
- He, J., Yao, X.: A study of drift analysis for estimating computation time of evolutionary algorithms. *Natural Computing*, **3**(1) (2004)
- Jägersküpfer, J.: Algorithmic analysis of a basic evolutionary algorithm for continuous optimization. *Theor. Comput. Sci.* **39**(3), 329–347 (2007)
- Jansen, T., De Jong, K.A., Wegener, I.: On the choice of the offspring population size in evolutionary algorithms. *Evol. Comput.* **13**(4), 413–440 (2005)
- Jansen, T., Sudholt, D.: Analysis of an asymmetric mutation operator. *Evol. Comput.* **18**, 1–26 (2010)
- Johnson, D.S., Papadimitriou, C.H., Yannakakis, M.: How easy is local search? *J. Comput. Syst. Sci.* **37**(1), 79–100 (1988)
- Kennedy, J., Eberhart, R.C.: *Swarm Intelligence*. Morgan Kaufmann, San Mateo (2001)
- Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. *Science* **220**(4598), 671–680 (1983)
- Larrañaga, P., Lozano, J.A.: *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic, Dordrecht (2002)
- Lehre, P.K., Witt, C.: Black-box search by unbiased variation. In: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation (GECCO'10), pp. 1441–1448. ACM, New York (2010)
- Motwani, R., Raghavan, P.: *Randomized Algorithms*. Cambridge University Press, Cambridge (1995)
- Neumann, F., Wegener, I.: Randomized local search, evolutionary algorithms, and the minimum spanning tree problem. *Theor. Comput. Sci.* **378**(1), 32–40 (2007)
- Neumann, F., Witt, C.: Ant colony optimization and the minimum spanning tree problem. In: Proceedings of Learning and Intelligent Optimization (LION'08), pp. 153–166 (2008)
- Neumann, F., Witt, C.: *Bioinspired Computation in Combinatorial Optimization: Algorithms and Their Computational Complexity*, 1st edn. Springer, New York (2010)
- Oliveto, P.S., Lehre, P.K., Neumann, F.: Theoretical analysis of rank-based mutation—combining exploration and exploitation. In: Proceedings of the 10th IEEE Congress on Evolutionary Computation (CEC'09), pp. 1455–1462. IEEE, New York (2009)
- Rowe, J.E., Vose, M.D.: Unbiased black box search algorithms. In: Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation (GECCO'11), pp. 2035–2042. ACM, New York (2011)

28. Rowe, J.E., Vose, M.D., Wright, A.H.: Neighborhood graphs and symmetric genetic operators. In: Proceedings of the 9th International Workshop on Foundations of Genetic Algorithms (FOGA'07). LNCS, vol. 4436, pp. 110–122 (2007)
29. Sudholt, D.: General lower bounds for the running time of evolutionary algorithms. In: Proceedings of the 11th International Conference on Parallel Problem Solving from Nature (PPSN'10), pp. 124–133. Springer, Berlin (2010)
30. Teytaud, O., Gelly, S.: General lower bounds for evolutionary algorithms. In: Proceedings of the 9th International Conference on Parallel Problem Solving from Nature (PPSN'06). LNCS, vol. 4193, pp. 21–31. Springer, Berlin (2006)
31. Teytaud, O., Gelly, S., Mary, J.: On the ultimate convergence rates for isotropic algorithms and the best choices among various forms of isotropy. In: Proceedings of the 9th International Conference on Parallel Problem Solving from Nature (PPSN'06). LNCS, vol. 4193, pp. 32–41. Springer, Berlin (2006)
32. Wegener, I.: Methods for the analysis of evolutionary algorithms on pseudo-Boolean functions. In: Sarker, R., Mohammadian, M., Yao, X. (eds.) Evolutionary Optimization, pp. 349–369. Kluwer Academic, Dordrecht (2002)
33. Wegener, I., Witt, C.: On the optimization of monotone polynomials by simple randomized search heuristics. *Comb. Probab. Comput.* **14**(1), 225–247 (2005)
34. Witt, C.: Runtime analysis of the  $(\mu + 1)$  EA on simple pseudo-Boolean functions. *Evol. Comput.* **14**(1), 65–86 (2006)
35. Zarges, C.: Theoretical foundations of artificial immune systems. PhD thesis, Technische Universität Dortmund (2011)