



Improved runtime results for simple randomised search heuristics on linear functions with a uniform constraint

Neumann, Frank; Pourhassan, Mojgan; Witt, Carsten

Published in:
Proceedings of the 2019 Genetic and Evolutionary Computation Conference

Link to article, DOI:
[10.1145/3321707.3321722](https://doi.org/10.1145/3321707.3321722)

Publication date:
2019

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Neumann, F., Pourhassan, M., & Witt, C. (2019). Improved runtime results for simple randomised search heuristics on linear functions with a uniform constraint. In *Proceedings of the 2019 Genetic and Evolutionary Computation Conference* (pp. 1506-1514). Association for Computing Machinery.
<https://doi.org/10.1145/3321707.3321722>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Improved Runtime Results for Simple Randomised Search Heuristics on Linear Functions with a Uniform Constraint

Frank Neumann

University of Adelaide

Adelaide, SA, Australia

frank.neumann@adelaide.edu.au

Mojgan Pourhassan

University of Adelaide

Adelaide, SA, Australia

mojgan.pourhassan@adelaide.edu.au

Carsten Witt

Technical University of Denmark

Kgs. Lyngby, Denmark

cawi@dtu.dk

ABSTRACT

In the last decade remarkable progress has been made in development of suitable proof techniques for analysing randomised search heuristics. The theoretical investigation of these algorithms on classes of functions is essential to the understanding of the underlying stochastic process. Linear functions have been traditionally studied in this area resulting in tight bounds on the expected optimisation time of simple randomised search algorithms for this class of problems. Recently, the constrained version of this problem has gained attention and some theoretical results have also been obtained on this class of problems. In this paper we study the class of linear functions under uniform constraint and investigate the expected optimisation time of Randomised Local Search (RLS) and a simple evolutionary algorithm called (1+1) EA. We prove a tight bound of $\Theta(n^2)$ for RLS and improve the previously best known bound of (1+1) EA from $O(n^2 \log(Bw_{\max}))$ to $O(n^2 \log B)$ in expectation and to $O(n^2 \log n)$ with high probability, where w_{\max} and B are the maximum weight of the linear objective function and the bound of the uniform constraint, respectively.

CCS CONCEPTS

- Theory of computation → Theory of randomized search heuristics;

KEYWORDS

randomised search heuristics, (1+1) EA, linear functions, constraints, runtime analysis

ACM Reference Format:

Frank Neumann, Mojgan Pourhassan, and Carsten Witt. 2019. Improved Runtime Results for Simple Randomised Search Heuristics on Linear Functions with a Uniform Constraint. In *Genetic and Evolutionary Computation Conference (GECCO '19), July 13–17, 2019, Prague, Czech Republic*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3321707.3321722>

1 INTRODUCTION

Randomised search heuristics, such as evolutionary computing techniques and randomised local search algorithms have been widely

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO '19, July 13–17, 2019, Prague, Czech Republic

© 2019 Copyright held by the owner/author(s). Publication rights licensed to the Association for Computing Machinery.
ACM ISBN 978-1-4503-6111-8/19/07...\$15.00
<https://doi.org/10.1145/3321707.3321722>

used in real world applications that involve optimisation. Over the last decade a lot of progress has been obtained in understanding the running time behaviour of these algorithms, which give us insights on the underlying stochastic process, particularly for classes of optimisation problems.

One of the classes of problems which has been studied for a simple evolutionary algorithm, called (1+1) EA, is the class of linear pseudo-boolean functions [4, 9, 12–14, 23]. The problem is to optimise a linear function of n Boolean variables. An upper bound of $O(n \log n)$ is first obtained for the optimisation time of (1+1) EA on this problem by Droste, Jansen and Wegener in [9], where the presented proof is highly technical. Later, using the analytic framework of drift analysis [11], He and Yao presented a simplified proof for the same upper bound of $O(n \log n)$ [12]. Another major improvement was made in [13, 14], where the first precise analysis is presented for the optimisation time of the problem. Using a framework for the analysis of multiplicative drift [5], Doerr, Johannsen and Winzen improved the precise upper bound result to the bound $(1.39 + o(1))en \ln n$ [4]. This bound was again further improved by Witt [23] to $en \ln n + O(n)$, who used adaptive drift [2, 3] based on a novel potential function.

The mentioned results consider the problem without any constraints. However, the class of linear pseudo-boolean functions has also been recently studied under linear constraints [10]. The problem of optimising a linear function under a linear constraint means that the search space is split by a hyperplane and only the points in one of the half spaces are considered feasible. This problem is equivalent to the well-known knapsack problem in the Boolean domain. One of the linear constraints that is studied in [10], is the uniform constraint, in which the constraint is given by OneMax; hence, restricting the number of 1-bits in the string. Denoting the bound on the number of 1-bits by B , the authors of that work have conjectured a general upper bound of $O(n^2)$ for all linear functions, independently of B . However, their analysis only proves a general upper bound of $O(n^2 \log(Bw_{\max}))$ for this setting, where w_{\max} is the largest weight in the objective function.

In this paper, we study two randomised search heuristics, RLS and (1+1) EA, and analyse the expected optimisation time of these algorithms on the linear function problem under a uniform constraint. We prove that an upper bound of $O(n^2)$ holds for RLS and we improve the current upper bound of $O(n^2 \log(Bw_{\max}))$ to $O(n^2 \log B)$ for (1+1) EA.

The problem of optimising a linear function under a uniform constraint can be seen as a simplification of the classical minimum spanning tree problem. The minimum spanning tree problem has been studied quite extensively in the area of randomised search heuristics. Neumann and Wegener [19] have shown an upper bound

Algorithm 1 (1+1) EA

```

 $t := 0.$ 
Choose uniformly at random  $x_0 \in \{0, 1\}^n$ .
repeat
  Create  $x'$  by flipping each bit in  $x_t$  independently with probability  $1/n$ .
   $x_{t+1} := x'$  if  $f(x') \leq f(x_t)$ , and  $x_{t+1} := x_t$  otherwise.
   $t := t + 1$ .
until some stopping criterion is fulfilled.

```

of $O(m^2(\log n + \log w_{\max}))$, where n is the number of nodes, m is the number of edges and w_{\max} is the largest edge weight of the given input graph. These results have been improved for special classes of graphs [24] and edge weights [20]. However, it still remains an open question whether an upper bound of $O(m^2 \log n)$ can be achieved for (1+1) EA on any graph.

The investigations in this paper are on a simpler problem, but we are hopeful that the provided techniques and insights will be helpful to achieve an upper bound of $O(m^2 \log n)$ of (1+1) EA on the minimum spanning tree problem. Many other analyses of evolutionary algorithms also contain the largest input weight in the obtained runtime bounds and getting strong results independent of this parameter poses a significant challenge for many problems where input weights might be exponential. This includes many results using multiplicative drift analysis when dealing with exponentially large weights and using the given fitness functions as the potential/drift function [6].

This paper is structured as follows. Section 2 includes the definition of the investigated algorithms and the analytical tools that we are going to use in the paper. Section 3 explains the studied problem, as well as the notations that we use in this paper. In Sections 4 and 5, respectively, we present the analysis for RLS and (1+1) EA, and finally, we finish in Section 6 with some conclusions.

2 PRELIMINARIES

We consider two classical randomised search heuristics called RLS and (1+1) EA, see Algorithms 1 and 2, which are intensively studied in the theory of randomised search heuristics [1, 15]. The (1+1) EA is a globally searching hill-climber, whereas RLS samples from a neighbourhood of size at most 2. Note that for RLS, steps that change two bits are crucial when the current search point has a tight constraint but is not the optimum yet.

The running time (synonymously, optimisation time) of the algorithms is defined as the random number of iterations until an optimal search point has been sampled. Denoting this number by a random variable T , in this paper we analyse the expected value of T , $E(T)$, for both studied algorithms.

In our analysis for (1+1) EA, we use two important drift theorems, which we list in this section in Theorem 2.1 and 2.2. The variable drift theorem (Theorem 2.1) was independently proposed in [16, 18] and generalised in [21]. The multiplicative drift theorem (Theorem 2.2) is due to Doerr et al. [6] and was enhanced with tail bounds by Doerr and Goldberg [3]. Both theorems are formulated in a unified and slightly generalised manner here. The formulation in terms of an arbitrary stochastic process can also be found in

Algorithm 2 Randomised Local Search (RLS)

```

 $t := 0.$ 
Choose uniformly at random  $x_0 \in \{0, 1\}^n$ .
repeat
  Choose  $b \in \{1, 2\}$  uniformly. Create  $x'$  by flipping  $b$  bits in  $x_t$  chosen uniformly at random.
   $x_{t+1} := x'$  if  $f(x') \leq f(x_t)$ , and  $x_{t+1} := x_t$  otherwise.
   $t := t + 1$ .
until some stopping criterion is fulfilled.

```

[17]. The adaptation of the multiplicative drift theorem to arbitrary positive s_{\min} -values has first been stated in Doerr et al. [6].

THEOREM 2.1 (VARIABLE DRIFT, CF. [16, 21]). *Let $(X_t)_{t \geq 0}$, be a stochastic process, adapted to a filtration \mathcal{F}_t , over some state space $S \subseteq \{0\} \cup [s_{\min}, s_{\max}]$, where $s_{\min} > 0$. Furthermore, let $T := \min\{t \mid X_t = 0\}$ be the first hitting time of state 0. Suppose that there exists a monotonically increasing function $h: [s_{\min}, s_{\max}] \rightarrow \mathbb{R}^+$ such that $1/h$ is integrable, and for all $t < T$*

$$E(X_t - X_{t+1} \mid \mathcal{F}_t) \geq h(X_t).$$

Then,

$$E(T \mid \mathcal{F}_0) \leq \frac{s_{\min}}{h(s_{\min})} + \int_{s_{\min}}^{X_0} \frac{1}{h(s)} ds.$$

THEOREM 2.2 (MULTIPLICATIVE DRIFT, CF. [3, 6]). *Let $(X_t)_{t \geq 0}$, be a stochastic process, adapted to a filtration \mathcal{F}_t , over some state space $S \subseteq \{0\} \cup [s_{\min}, s_{\max}]$, where $s_{\min} > 0$. Suppose that there exists a $\delta > 0$ such that for all $t \geq 0$*

$$E(X_t - X_{t+1} \mid \mathcal{F}_t) \geq \delta X_t.$$

Then it holds for the first hitting time $T := \min\{t \mid X_t = 0\}$ that

$$E(T \mid \mathcal{F}_0) \leq \frac{\ln(X_0/s_{\min}) + 1}{\delta}.$$

Moreover, $\Pr(T > (\ln(X_0/s_{\min}) + r)/\delta) \leq e^{-r}$ for any $r > 0$.

Finally, in our analysis, we will use the following simple lemma dealing with convexity.

LEMMA 2.3. *Let $a_1, \dots, a_B \geq 0$ and $C > 1$. Then*

$$(a_1 + \dots + a_B)^C \leq B^{C-1} \left((a_1)^C + \dots + (a_B)^C \right).$$

Proof. We write

$$(a_1 + \dots + a_B)^C = B^C \left(\frac{a_1}{B} + \dots + \frac{a_B}{B} \right)^C,$$

and interpret the expression in parentheses as a linear combination of the B numbers with coefficient $1/B$ each. Applying Jensen's inequality, we have

$$\left(\frac{a_1}{B} + \dots + \frac{a_B}{B} \right)^C \leq \frac{a_1^C}{B} + \dots + \frac{a_B^C}{B},$$

which, after multiplying with B^C , gives the desired result. \square

Notation. Throughout this paper, for natural numbers n we write $[n] := \{1, \dots, n\}$.

3 SCENARIO

In this paper we analyse the expected optimisation time of RLS and (1+1) EA and consider an optimisation problem with a linear objective function under a uniform constraint. In contrast to earlier work in this area [10], we assume that the objective function has to be *minimised* since this perspective more naturally coincides with the minimisation of the distance to the target 0 that is implicit in the drift theorems (Theorems 2.1 and 2.2). The upper bounds on the optimisation time obtained for RLS in Section 4 and for (1+1) EA in Section 5, respectively, would equally hold if we adopted maximisation in the same way as in the previous work. See Section 5.3 for more discussion about assuming a minimisation problem or a maximisation problem.

Formally, throughout this paper, we consider the search space $\{0, 1\}^n$ of all bit strings $x = x_n x_{n-1} \cdots x_1$, and the goal is to *minimise* the objective function of

$$f_{\text{obj}}(x) = \sum_{i=1}^n w_i x_i,$$

where $w_n \geq \cdots \geq w_1$ are positive real weights, under the uniform constraint

$$x_1 + \cdots + x_n \geq B$$

for some $B \in \{1, \dots, n\}$. We are excluding $B = 0$, as it is equivalent to having no constraints. Note that we follow common conventions in the analysis of linear functions [4, 23] by writing down search points in the order $x_n \dots x_1$, i.e., most significant bit first. Therefore, an index i is called *left* of index $j \neq i$ if $i > j$ and *right* of j otherwise.

A search point is *optimal* if it minimises f_{obj} and is placed in the feasible region, i.e., the part of the search space where the constraint is satisfied. Moreover, we say that a search point is *tight* (in the constraint) if $x_1 + \cdots + x_n = B$.

In Algorithms 1 and 2, x denotes the best search point found so far, and x' is the new offspring, which replaces x if it is at least as good as it with respect to a fitness function f that we define as follows. We aim to handle the constraint of the problem by setting a penalty for the violation. Therefore, we define the fitness function below, to be used by the algorithms.

$$f(x) = f_{\text{obj}}(x) + \max\{0, (B - b(x))\} \cdot (nw_{\max} + 1)$$

where $w_{\max} = w_n$ is the maximal weight, and $b(x) = \sum_{i=1}^n x_i$ is the number of ones in the bit string x , which we also refer to as the *b-value* of x . For feasible search points we have $b(x) \geq B$, which implies that $\max\{0, (B - b(x))\} = 0$. Therefore, the penalty of $(B - b(x)) \cdot (nw_{\max} + 1)$ is applied to search points that are infeasible, making the value of this fitness function larger than that of any feasible search points. Note that with this definition of the fitness function, the search in infeasible region is also guided to the feasible region, because as the extent of the constraint violation is reduced the penalty also decreases.

We first find a tight bound on the expected optimisation time of RLS on this problem in Section 4, and then focus on the challenging analysis of (1+1) EA in the rest of the paper. Lemma 4.1, which is presented in Section 4 holds for (1+1) EA as well as RLS, and is used in analysis of both algorithms (Section 4 and Section 5).

4 ANALYSIS OF RLS

In Theorem 4.2, we prove that RLS (Algorithm 2) optimises the linear function problem under a uniform constraint in expected time $O(n^2)$. In Theorem 4.3 we also prove that this bound is tight.

We start with the following lemma, which proves that a feasible search point is sampled by the studied algorithm in $O(n \log(n/(n - B)))$. This lemma holds for (1+1) EA as well, and is also used in analysis of Section 5. The proof of this lemma is similar to the proof of Lemma 7 in [10] in which a maximisation problem for a linear function under uniform constraint is considered. Here we adapt the proof to match the minimisation problem.

LEMMA 4.1. *Starting with an arbitrary initial solution, the expected time until RLS or (1+1) EA obtain a feasible solution is $O(n \log(n/(n - B)))$.*

Proof. Recall that we denote by $b(x)$ the number of 1-bits in a search point x . Due to the definition of the fitness function f , in the infeasible region, a search point x with a larger $b(x)$ is always preferred to a search point with a smaller b -value. Therefore, the problem can be seen as maximising $b(x)$ until reaching $b(x) \geq B$, where the initial solution may have a b -value of 0. We consider the potential function

$$g(x) = \begin{cases} n - b(x), & \text{if } b(x) < B, \\ 0, & \text{otherwise.} \end{cases}$$

for which the initial value is at most n and the minimum value before reaching 0 is $n - B + 1$. The value of this function is never increased during the process of RLS or (1+1) EA, as larger b -values are always preferred to smaller b -values before reaching $g(x) = 0$. We find the drift on the value of $g(x_t)$ for RLS, where x_t is the search point of the algorithm at step t , as

$$\mathbb{E}(g(x_t) - g(x_{t+1}) \mid g(x_t); g(x_t) > 0) \geq \frac{n - b(x_t)}{2n} = \frac{g(x_t)}{2n}$$

since RLS performs a 1-bit flip with probability $1/2$ and flips a 0-bit with probability $(n - b(x_t))/n$, improving g by 1. A similar drift of

$$\mathbb{E}(g(x_t) - g(x_{t+1}) \mid g(x_t); g(x_t) > 0) \geq \frac{g(x_t)}{en}$$

is obtained for (1+1) EA, in which the probability of flipping one 0-bit and no other bits is $\frac{n - b(x_t)}{n} \cdot (1 - \frac{1}{n})^{n-1} \geq \frac{g(x_t)}{en}$.

Using the multiplicative drift theorem (Theorem 2.2) with $\delta = 1/en$, $X_0 \leq n$ and $s_{\min} = n - B + 1$ we find that the expected time until reaching a feasible solution is upper bounded by

$$\frac{\ln(n/(n - B + 1)) + 1}{1/(en)} = O\left(\ln\left(\frac{n}{n - B}\right)\right).$$

□

THEOREM 4.2. *Starting with an arbitrary initial solution, the expected optimisation time of RLS on linear functions with a uniform constraint is $O(n^2)$.*

Proof. Due to Lemma 4.1, RLS finds a feasible solution in expected time $O(n \log(n/(n - B)))$. Also, since all feasible solutions have smaller fitness values than infeasible solutions, the algorithm does not switch back to the infeasible region again. Moreover, note that once a feasible solution has been found for the first time, the number of ones in the solution cannot be increased. This is due to the fact

that the penalty is 0 and all 1-bit flips flipping a 0 increase the fitness. Also, all 2-bit flips that increase the number of ones (flipping two zeros) increase the fitness as well.

We split the analysis of the algorithm after obtaining a feasible solution into two phases. In the first phase, the algorithm starts with a solution x with $b(x) > B$ and obtains a solution with exactly B 1-bits ($b(x) = B$). Then the second phase starts, during which the number of 1-bits of the solution does not change, because both 1-bit flips and 2-bit flips that change the number of ones increase the fitness. If the first feasible solution that is obtained by the algorithm has $b(x) = B$, then we do not have a first phase. We first analyse the expected time until the first phase ends, then we focus on the second phase.

In the first phase, the algorithm starts with a solution x with $b(x) > B$. In this situation, as explained above, $b(x)$ does not increase. Moreover, a 1-bit flip that flips a 1 to 0, which happens with probability $b(x)/(2n)$, is always accepted because it decreases the fitness, while not violating the constraint yet. By defining a potential function $g(x)$ as

$$g(x) = \begin{cases} b(x), & \text{if } b(x) > B. \\ 0, & \text{otherwise.} \end{cases}$$

and using multiplicative drift theorem with $\delta = 1/2n$, $X_0 \leq n$ and $s_{\min} = B + 1$, we find the expected time of $O(n \log(n/B))$ until a solution with $g(x) = 0$ is found, which implies $b(x) = B$.

Now we analyse the second phase. Having obtained a solution with exactly B ones, only 2-bit flips flipping a zero and a one are accepted. Let r be the number of bits of weight w_B among w_B, \dots, w_1 , i.e. $r = |\{i \mid w_i = w_B, 1 \leq i \leq B\}|$. An optimal solution contains all weights of weight less than w_B and exactly r weights of weight w_B .

Let x be the current solution and $s(x) = \max\{0, r - |\{i \mid w_i = w_B \wedge x_i = 1\}|\}$ be the number of 1-bits of weight w_B missing in x . Furthermore, let

$$t(x) = |\{i \mid w_i < w_B \wedge x_i = 0\}|$$

be the number of 1-bits of weight less than w_B missing in x .

We denote by

$$k = s(x) + t(x)$$

the number of weights that are missing in the weight profile of the current solution x compared to an arbitrary optimal solution.

As there are exactly B 1-bits in the current solution x , it implies that there are exactly

$$k = |\{i \mid w_i > w_B \wedge x_i = 1\}| + \max\{0, |\{i \mid w_i = w_B \wedge x_i = 1\}|-r\}$$

weights chosen in x that do not belong to an optimal weight profile. Note that for a given solution x

$$r - |\{i \mid w_i = w_B \wedge x_i = 1\}|$$

is a fixed value which is greater than 0 if 1-bits of weight w_B are missing and less than 0 if there are too many 1-bits of weight w_B .

This implies that there are at k 1-bits which can be swapped with an arbitrary 0-bit of the missing k weights in order to reduce k . Hence, the probability of swapping a 1-bit with a 0-bit of the missing weights is at least $\frac{k^2}{2n^2}$ and the expected waiting time for this event is bounded from above by $2n^2/k^2$. Since k cannot increase, it suffices to sum up these expected waiting times following the

idea of fitness-based partitions [22]. Hence, the expected time until reaching $k = 0$ is

$$\sum_{k=1}^B (2n^2/k^2) = O(n^2),$$

which completes the proof. \square

We now show that the previous bound is asymptotically tight.

THEOREM 4.3. *There is a linear function f and a bound B such that, starting with a uniformly random initial solution, the expected optimisation time of RLS on f under uniform constraint B is $\Omega(n^2)$.*

Proof. The same lower bound is proved for (1+1) EA in Theorem 10 of [10]. Since RLS does not flip more than 2 bits at each step, the proof of this theorem is simpler. We use a function f that is similar to the function that is used in [10] and is adapted for a minimisation problem. We define f as

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

where ε is an arbitrary positive real number. Since the weights that are assigned to the first B bits are smaller than the weights of other bits, the optimal solution is selecting the first B bits. We prove that with $B = n/4$, the expected optimisation time of RLS is lower bounded by $\Omega(n^2)$.

We denote the Hamming distance of a solution x to the optimal solution by $d_H(x)$. By Chernoff bounds the initial solution has at least $n/3$ 1-bits with probability exponentially close to 1, which implies a Hamming distance of at least $n/12$ to the optimal solution. Since RLS can only decrease the Hamming distance by one or two at each step, in order to reach the optimal solution, a solution x has to be obtained at some point such that $2 \leq d_H(x) \leq 3$. We investigate the process based on the number of 1-bits of solution x , which we denote by $|x|_1$. Since the initial solution is feasible with probability exponentially close to 1, we either have $|x|_1 = B$ or $|x|_1 > B$.

If $|x|_1 = B$, then $d_H(x) = 2$ and x can only have one 0-bit among the first B bits and one 1-bit among other bits. In this case only a swap on the two misplaced bits can improve the fitness, the probability of which to happen is at most $1/n^2$; hence, the waiting time is $\Omega(n^2)$ and the theorem follows.

If $|x|_1 > B$, then flipping any of the 1-bits improves the fitness. Since there are more than $n/4$ 1-bits in the solution, the probability of decreasing the number of 1-bits is at least $1/8$ at each time step of RLS. Furthermore, the number of 0-bits does not decrease by RLS due to the fitness function. Using a drift argument on $|x|_1 - B$, we find that in expected constant time (at most $\frac{3}{1/8}$) a solution x' is obtained such that $|x'|_1 = B$. This implies that in a phase of $\log n$ steps, with probability $1 - o(1)$ the solution x' is obtained. If x' is not optimum, then we have to swap at least two bits and the theorem follows as above. What remains is to show that x' is not optimum with probability $1 - o(1)$. Since $d_H(x) \leq 3$, the probability of flipping a one-bit from x that is outside the first B positions is at most $3/n$ at each step. Therefore, with probability at least $1 - (1 - (1 - 1/n)^{\log n})^3 = 1 - o(1)$ at least one of these bits does not flip in a phase of $\log n$ steps; hence, x' is not the optimal solution with probability $1 - o(1)$, which completes the proof. \square

5 ANALYSIS OF (1+1) EA

In this section we analyse the expected optimisation time of (1+1) EA for the linear function under a uniform constraint. In Section 5.1 we present the statement of our results, and then in the following section we prove the statement.

5.1 Main Theorem

For a linear function under uniform constraint of B , we aim to prove that (1+1) EA finds an optimal solution in expected time $O(n^2 \log^+ B)$, where $\log^+(x) := \max\{1, \log x\}$. Since Lemma 4.1 proves that a feasible solution is obtained by (1+1) EA in expected time $O(n \log(n/(n - B)))$ and this upper bound is asymptotically smaller than $O(n^2 \log^+ B)$, we only focus on the analysis of the algorithm after finding a feasible solution. The main theorem that we prove in this section is stated below.

THEOREM 5.1. *Given an arbitrary linear function under a uniform constraint $x_1 + \dots + x_n \geq B$ for $B \in \{1, \dots, n\}$, the expected optimisation time of the (1+1) EA is upper bounded by $O(n^2 \log^+ B)$, where $\log^+(x) := \max\{1, \log x\}$. Also, the time is $O(n^2 \log n)$ with probability $1 - O(n^{-c})$ for any constant $c > 0$.*

5.2 Adaptive Drift Analysis

To prove Theorem 5.1, we conduct an adaptive drift analysis, where the underlying potential function $g(x)$, to be minimised, depends on both the weights (w_1, \dots, w_n) of the linear function and the constraint value B . The exact definition of the potential function is to some extent inspired by the techniques developed in Witt [23] and further applied in Doerr and Pohl [7] and Doerr et al. [8]. However, as these papers are concerned with unconstrained problems only, additional effort has been made to transfer these techniques to our scenario.

Once having defined the potential function, the idea is to analyse the potential $X_t := g(x^{(t)})$ of the random search point $x^{(t)}$ maintained by the (1+1) EA on f at time t . We bound its expected one-step change $E(X_t - X_{t+1} | X_t)$, i.e., the expected decrease of the potential function from time t to time $t + 1$. Then we use this bound in the drift argument that proves the main theorem.

The following lemma (Lemma 5.2) states this bound as well as a bound on the maximum value of the potential function, which will be required in the drift theorems. We define $g(x)$ later in Definition 5.3, and prove the statements of Lemma 5.2 for this function afterwards. We first bring the statement of this lemma and show how it can be used to prove Theorem 5.1.

LEMMA 5.2. *Considering a random variable $X_t = g(x^{(t)})$, where the function g is given in Definition 5.3 and $x^{(t)}$ is the random search point of (1+1) EA at time t , for all time steps t we have*

- (1) $E(X_t - X_{t+1} | X_t) \geq \frac{0.11X_t}{en^2} \max\{X_t^{1/7}/B^{1/7}, 1\}$.
- (2) $1 \leq X_t \leq n^8$ if $x^{(t)}$ is not optimal.

Deferring the definition of the potential function g and the proof of the previous lemma, we obtain our theorem.

Proof of Theorem 5.1. We apply the variable drift theorem (Theorem 2.1) given the statements of Lemma 5.2. Using that $X_t \geq 1 :=$

s_{\min} and $X_t \leq n^8$ as well as the drift bound

$$h(X_t) := \frac{0.11X_t}{en^2} \max\{X_t^{1/7}/B^{1/7}, 1\},$$

the expected optimisation time is bounded by

$$\begin{aligned} & \frac{s_{\min}}{h(s_{\min})} + \int_{s_{\min}}^{n^8} \frac{1}{h(x)} dx \\ &= O(n^2) + \frac{en^2}{0.11} \left(\int_1^B \frac{1}{x} dx + B^{1/7} \int_{B+1}^{n^8} \frac{1}{x^{8/7}} dx \right) \\ &= O(n^2) + O(n^2)(O(\log B) + O(1)) = O(n^2 \log^+ B), \end{aligned}$$

which completes the proof of the $O(n^2 \log^+ B)$ bound.

For the tail bound we use the multiplicative drift theorem (Theorem 2.2) with the simple bound $E(X_t - X_{t+1} | X_t) \geq \frac{0.11X_t}{en^2}$ of Lemma 5.2 along with $X_t \leq n^8$ that implies $\ln(X_t/s_{\min}) = O(\log n)$. Note that the theorem gives the upper bound $O(n^2 \log n)$ on the expected optimisation time so that the tail bound can be obtained by setting $r = c \ln n$. \square

In the following, we unroll the proofs of the drift statements. The proof of Lemma 5.2 relies on the analysis of the one-step drift of the potential function $g: \{0, 1\}^m \rightarrow \mathbb{R}$. We now introduce the setup required to define $g(x)$.

Definition 5.3. Let an arbitrary linear function $f = \sum_{i=1}^n w_i x_i$, where $w_n \geq \dots \geq w_1$, under uniform constraint $x_1 + \dots + x_n \geq B$ be given and let x_{opt} be the (not necessarily unique) optimal search point having one-bits at the B least significant positions only. Let $m := |\{w_{B+1}, \dots, w_n\}|$ be the number of distinct weights at the $n - B$ most significant positions and define $s(i) = \min\{j \mid |\{w_{B+1}, \dots, w_j\}| \geq i\}$, where $i \in \{1, \dots, m\}$, as the start of the block of indices having the i th largest weight as well as $s(m+1) := n+1$. Also, let $K_i := \{s(i), \dots, s(i+1)-1\}$ be the indices comprising the i th block of equal weights.

For $j \in \{B+1, \dots, n\}$, let

$$\gamma_j := (j - B)^7$$

and $\gamma_1 = \dots = \gamma_B := 0$. Based on this, define for all blocks $i \in [m]$

$$g_{s(i)} = \dots = g_{s(i+1)-1} := \min\{\gamma_{s(i)}, g_{s(i-1)} \cdot w_{s(i)}/w_{s(i-1)}\}$$

as well as $g_{B+1} := 1$, $g_1 = \dots = g_B := 0$, and $g(x) := \sum_{i=1}^n g_i x_i$. For any block $i \in \{1, \dots, m\}$, we also define:

- $\kappa(i) := \max\{j \leq i \mid g_{s(j)} = \gamma_{s(j)}\}$, the most significant block right of i (possibly i itself) capping according to the sequence γ_i ,
- $L(i) := \{m, \dots, \kappa(i)\}$, the block indices left of (and including) the block $\kappa(i)$,
- $R(i) := \{\kappa(i) - 1, \dots, 1\}$, the block indices right of block $\kappa(i)$.

Considering the original weights w_1, \dots, w_n in increasing order, the potential function assigns the same g -value to all indices within a block K_i of equal w -value. Note that blocks may be of size 1. We also observe that the weights of g can be equivalently defined as $g_j = \min\{\gamma_j, g_{j-1} \cdot w_j/w_{j-1}\}$ for $j \in \{B+1, \dots, n\}$.

The idea of the potential function is to cap the original weights at γ_i at the indices where the original weights increase too steeply and to rebuild their slope otherwise. In particular, we have $g_i \leq \gamma_i$

for all $i \in [n]$. The intuition is that the potential function will underestimate the progress made at blocks being at least as significant as $\kappa(i)$, i.e., the blocks in $L(i)$. In all less significant blocks (those in $R(i)$), we will pessimistically assume that they contribute a loss, and the choice of $\kappa(i)$ guarantees that this loss is overestimated.

Also, the potential function assigns a value of zero to one-bits at the rightmost B positions since these correspond to an optimal setting.

As mentioned above, we will analyse the stochastic process $(X_t)_{t \geq 0}$ where $X_t = g(x^{(t)})$ for all t , and define $\Delta_t := X_t - X_{t+1}$. Recall that we are interested in the first point in time t where $X_t = 0$ holds. The one-step drift $E(\Delta_t | X_t)$ of the potential function will be worked out conditioned on certain events depending on two flipping bits. The following notions prepare the definition of these events.

Definition 5.4. Given $x^{(t)} \in \{0, 1\}^n$, denote by x' the random search point created by mutation of $x^{(t)}$ (before selection). We define

- $I := \{i \in [n] \mid x_i^{(t)} = 1\}$ the one-bits in $x^{(t)}$,
- $I^* := \{i \in I \mid x'_i = 0\}$ the one-bits flipping to 0,
- $Z := \{i \in [n] \mid x_i^{(t)} = 0\}$ the zero-bits in $x^{(t)}$,
- $Z^* := \{i \in Z \mid x'_i = 1\}$ the zero-bits flipping to 1,
- $\sigma_i := |I^* \cap K_i| - |Z^* \cap K_i|$ the surplus of flipping one-bits within block K_i , where $i \in [m]$.

Note that the random sets I^* and Z^* are disjoint and that the remaining bits in $[n]$ contribute nothing to the Δ_t -value.

Obviously, for $\Delta_t \neq 0$ it is necessary that $x^{(t+1)} \neq x^{(t)}$. We fix an arbitrary search point $x^{(t)}$ and let A be the event that $x^{(t+1)} \neq x^{(t)}$. Then event A requires that

$$I^* \neq \emptyset \text{ and } \sum_{j \in I^*} w_j - \sum_{j \in Z^*} w_j \geq 0.$$

To simplify the analysis of blocks of equal weights, we from now on use the equivalence

$$\sum_{j \in I^*} w_j - \sum_{j \in Z^*} w_j = \sum_{i=1}^m \sigma_i w_{s(i)}$$

Hence, for A to occur it is necessary that

$$\sum_{i \mid \sigma_i > 0} |\sigma_i| w_{s(i)} - \sum_{i \mid \sigma_i < 0 \wedge i \geq k} |\sigma_i| w_{s(i)} \geq 0,$$

for arbitrary $k \in [m]$ since we only ignore the loss due to the bits right of block k . In the following, $k = \kappa(i)$ will be used where i is the leftmost block such that $\sigma_i > 0$.

We now decompose the event A according to two indices $i \in [m]$, $\ell \in [n]$, where i relates to the leftmost block that flips more ones than zeros, and ℓ to the leftmost flipping one-bit from block i .

Definition 5.5. The event $A_{i,\ell}$, where $i \in [m]$ and $\ell \in [n]$, occurs iff the following conditions hold simultaneously.

- (1) $I^* \neq \emptyset$.
- (2) $i := \max\{i \mid \sigma_i > 0\}$.
- (3) $\ell = \max(I^* \cap K_i)$
- (4) $\sum_{j \mid \sigma_j > 0} |\sigma_j| w_{s(j)} - \sum_{j \mid \sigma_j < 0 \wedge j \geq \kappa(i)} |\sigma_j| w_{s(j)} \geq 0$.
- (5) A feasible search point is obtained by flipping the bits from $I^* \cup Z^*$ in $x^{(t)}$.

Obviously, the events $A_{i,\ell}$ are mutually disjoint. Since each accepted, non-copying mutation must flip at least one one-bit, the union of the events $A_{i,\ell}$ is a superset of A (in other words, is necessary for A). The key inequality used to bound the one-step drift is stated in the following lemma.

LEMMA 5.6. $E(\Delta_t | A_{i,\ell}) \geq 0.11g_{s(i)}$ for all $i \in [m]$ and $\ell \in K_i$ such that $\Pr(A_{i,\ell}) > 0$, and all $t \geq 0$.

Before we prove Lemma 5.6, let us show how it can be used to prove Lemma 5.2.

Proof of Lemma 5.2. We still fix an arbitrary search point $x^{(t)}$, denote by $X_t = g(x^{(t)})$ its potential and investigate the following step. As observed above, in the step the potential remains either unchanged or a certain event $A_{i,\ell}$ occurs. The total drift can then be expressed as

$$E(X_t - X_{t+1} | X_t) = \sum_{i \in [m], \ell \in K_i, \Pr(A_{i,\ell}) > 0} E(\Delta_t | A_{i,\ell}) \cdot \Pr(A_{i,\ell}).$$

Using Lemma 5.6, the last expression is at least

$$\sum_{i \in [m], \ell \in K_i, \Pr(A_{i,\ell}) > 0} 0.11g_{s(i)} \Pr(A_{i,\ell}), \quad (1)$$

so we have to bound $\Pr(A_{i,\ell})$ from below for those events that are possible.

If $\Pr(A_{i,\ell}) > 0$ then there is a one-bit at position ℓ . If the current search point is not tight, already flipping bit ℓ only is accepted. Hence, we now pessimistically assume that the constraint is tight so that there are $B-1$ other one-bits in $x^{(t)}$. Considering a mutation that flips bit ℓ the mutation is accepted if it flips a zero-bit right of ℓ and does not flip any further bits. Even if all B one-bits are right of (and including) bit ℓ , there are at least $\ell - B$ zero-bits right of ℓ . Noting that the probability of not flipping $n-2$ bits is $(1-1/n)^{n-2}$, we conclude that

$$\Pr(A_{i,\ell}) \geq \frac{\ell - B}{n^2} \left(1 - \frac{1}{n}\right)^{n-2} \geq \frac{\ell - B}{en^2} \quad (2)$$

if $A_{i,\ell}$ is possible. We will now relate this bound to the factor $g_{s(i)}$ appearing in (1). First of all, since ℓ appears in block i and all bits in a block have equal weight, we have $g_{s(i)} = g_\ell$. Next we note that $g_\ell \leq \gamma_\ell = (\ell - B)^7$ by Definition 5.3. Plugging this into Equation (2), we obtain (if $A_{i,\ell}$ is possible) that

$$\Pr(A_{i,\ell}) \geq \frac{(g_{s(i)})^{1/7}}{en^2} \quad (3)$$

and of course also $\Pr(A_{i,\ell}) \geq 1/(en^2)$ by just estimating $\ell - B \geq 1$ in (2).

Furthermore, the one-bits outside positions $1, \dots, B$ altogether make up the current g -value. Let \tilde{I} be the set of these bits. Since for each $i \in [m]$ there are $K_i \cap \tilde{I}$ disjoint events $A_{i,\ell}$ of probability at least $(g_{s(i)})^{1/7}/(en^2)$ each, namely one for each one-bit ℓ within

block i , we obtain by combining (1) and (3) that

$$\begin{aligned} & \mathbb{E}(X_t - X_{t+1} \mid X_t) \\ & \geq \sum_{i \in [m] \mid K_i \cap \tilde{I} \neq \emptyset} |K_i \cap \tilde{I}| 0.11 g_{s(i)} \Pr(A_{i,\ell}) \\ & \geq \sum_{i \in [m] \mid K_i \cap \tilde{I} \neq \emptyset} |K_i \cap \tilde{I}| \frac{0.11(g_{s(i)})^{8/7}}{en^2} \\ & \geq \frac{0.11(g(x^{(t)})^{8/7})}{B^{1/7}en^2} \end{aligned}$$

where the last inequality used the estimate

$$(a_1 + \dots + a_B)^{8/7} \leq B^{1/7}((a_1)^{8/7} + \dots + (a_B)^{8/7})$$

proved in Lemma 2.3. Using also the simple bound $\Pr(A_{i,\ell}) \geq 1/(en^2)$ and recalling that $X_t = g(x^{(t)})$, we altogether have

$$\mathbb{E}(X_t - X_{t+1} \mid X_t) \geq \frac{0.11X_t}{en^2} \max\left\{X_t^{1/7}/B^{1/7}, 1\right\}.$$

This proves the first statement of Lemma 5.2.

For the second statement of Lemma 5.2, we simply use that $g_i \leq i^7$, so for all $x^{(t)}$ it holds that $g(x^{(t)}) \leq \sum_{i=1}^n g_i \leq n \cdot n^7 = n^8$. Also, since $g_{B+1} = 1$ each non-optimal search point $x^{(t)}$ must satisfy $g(x^{(t)}) \geq 1$. \square

The still outstanding proof of Lemma 5.6 requires a careful analysis of the one-step drift, taking into account the specific structure of the drift function.

Proof of Lemma 5.6. Recall that we want to condition on the event $A_{i,\ell}$ (Definition 5.5), where i is the leftmost block flipping more ones than zeros. Moreover, recall the notions introduced in Definitions 5.3 and 5.4. Let

$$\begin{aligned} \Delta_L(i) &:= \left(\sum_{j \mid \sigma_j > 0} |\sigma_j| g_{s(j)} - \sum_{j \mid \sigma_j < 0 \wedge j \geq \kappa(i)} |\sigma_j| g_{s(j)} \right) \cdot \mathbb{1}_A, \\ \Delta_R(i) &:= \left(\sum_{j \mid \sigma_j > 0 \wedge j < \kappa(i)} |\sigma_j| g_{s(j)} \right) \cdot \mathbb{1}_A, \end{aligned}$$

where $\mathbb{1}_A$ denotes the indicator random variable of event A . Recall that $\Delta_t = 0$ if A does not occur. Otherwise, $\Delta_t = \sum_{j \mid \sigma_j > 0} |\sigma_j| g_{s(j)} - \sum_{j \mid \sigma_j < 0} |\sigma_j| g_{s(j)}$. Hence, we have $\Delta_t = (\Delta_L(i) - \Delta_R(i))$ for all $i \in [m]$. By linearity of expectation, we obtain

$$\mathbb{E}(\Delta_t \mid A_{i,\ell}) = \mathbb{E}(\Delta_L(i) \mid A_{i,\ell}) - \mathbb{E}(\Delta_R(i) \mid A_{i,\ell}). \quad (4)$$

We first show that $(\Delta_L(i) \mid A_{i,\ell})$ is a non-negative random variable, i. e., the probability of any negative outcome is 0. To prove this, assume that $A_{i,\ell}$ holds, which implies that no block left of i flips more ones than zeros.

We now inspect the relation between the weights of the original function and the potential function. Here we exploit that the ratio of g -values and w -values of two blocks $i > j$ is the same unless the weight of block i is capped by the minimum operator in the definition of $g_{s(i)}$ in Definition 5.3. Otherwise, the ratio may be smaller. Looking also into symmetrical cases, for any $i \in [m]$ we

obtain from Definition 5.3 that

$$\frac{g_{s(j)}}{g_{s(\kappa(i))}} = \frac{w_{s(j)}}{w_{s(\kappa(i))}} \text{ for } i \geq j \geq \kappa(i). \quad (5)$$

$$\frac{g_{s(j)}}{g_{s(\kappa(i))}} \leq \frac{w_{s(j)}}{w_{s(\kappa(i))}} \text{ for } j \geq \kappa(i) \quad (6)$$

$$\frac{g_{s(j)}}{g_{s(\kappa(i))}} \geq \frac{w_{s(j)}}{w_{s(\kappa(i))}} \text{ for } j < \kappa(i). \quad (7)$$

Hence,

$$\begin{aligned} (\Delta_L(i) \mid A_{i,\ell}) &= \left(\sum_{j \mid \sigma_j > 0} |\sigma_j| g_{s(j)} - \sum_{j \mid \sigma_j < 0 \wedge j \geq \kappa(i)} |\sigma_j| g_{s(j)} \right) \\ &\geq \left(\sum_{j \mid \sigma_j > 0} |\sigma_j| g_{s(\kappa(i))} \frac{w_{s(j)}}{w_{s(\kappa(i))}} - \sum_{j \mid \sigma_j < 0 \wedge j \geq \kappa(i)} |\sigma_j| g_{s(\kappa(i))} \frac{w_{s(j)}}{w_{s(\kappa(i))}} \right) \\ &\geq 0 \end{aligned}$$

where the first inequality uses (5)–(7) along with the fact that no block left of i has positive σ -value, and the last inequality holds by the fourth item from the definition of $A_{i,\ell}$ (Definition 5.5).

We note that according to the fifth item of Definition 5.5, this event may imply that a bit $j^* \in Z$ flips to 1 simultaneously with a one-bit in block i flipping to 0. This is the case if the constraint is tight in the search point $x^{(t)}$, which we again pessimistically assume to be the case (if $x^{(t)}$ had more than B one-bits, flipping only ℓ would already be accepted).

Now let $S_{i,\ell}$ be the event that the following three events happen simultaneously:

- (1) $|I^* \cup Z^* \cap K_j| = 0$ for all $j \in \{\kappa(i), \dots, m\} \setminus \{i\}$
- (2) $|I^* \cap K_i| = 1$ and $\ell \in I^* \cap K_i$,
- (3) $|Z^* \cap K_i| = 0$,

i. e., block i is the only one in $L(i)$ that contributes to Δ_L by flipping exactly one one-bit at position ℓ . We have

$$\begin{aligned} \mathbb{E}(\Delta_L(i) \mid A_{i,\ell}) &= \mathbb{E}(\Delta_L(i) \mid A_{i,\ell} \cap S_{i,\ell}) \cdot \Pr(S_{i,\ell} \mid A_{i,\ell}) \\ &\quad + \mathbb{E}(\Delta_L(i) \mid A_{i,\ell} \cap \bar{S}_{i,\ell}) \cdot \Pr(\bar{S}_{i,\ell} \mid A_{i,\ell}) \end{aligned}$$

by the law of total expectation. As the random variable $(\Delta_L(i) \mid A_{i,\ell})$ cannot have any negative outcomes, all these conditional expectations are non-negative as well. From (4) we thus derive

$$\mathbb{E}(\Delta_t \mid A_{i,\ell}) \geq \mathbb{E}(\Delta_L(i) \mid A_{i,\ell} \cap S_{i,\ell}) \cdot \Pr(S_{i,\ell} \mid A_{i,\ell}) - \mathbb{E}(\Delta_R(i) \mid A_{i,\ell}). \quad (8)$$

We will now bound the terms from (8) from below to obtain our result. For $(S_{i,\ell} \mid A_{i,\ell})$ to occur, it is sufficient that all bits in the blocks in $L(i)$ except the one-bit ℓ in block i and bit j^* do not flip (note that these bits flip since we condition on $A_{i,\ell}$). Consequently, $\Pr(S_{i,\ell} \mid A_{i,\ell}) \geq (1 - 1/n)^{n-2} \geq 1/e$. Moreover, since no zero-bits in $L(i)$ flip under $A_{i,\ell} \cap S_{i,\ell}$, j^* must be in a block in $R(i)$. Hence, $\mathbb{E}(\Delta_L(i) \mid A_{i,\ell} \cap S_{i,\ell}) \geq g_{s(i)}$. Altogether,

$$\mathbb{E}(\Delta_L(i) \mid A_{i,\ell} \cap S_{i,\ell}) \cdot \Pr(S_{i,\ell} \mid A_{i,\ell}) \geq \frac{g_{s(i)}}{e}. \quad (9)$$

Finally, we need a bound on $\mathbb{E}(\Delta_R(i) \mid A_{i,\ell})$, which is determined by the bits in $R(i)$ that flip to 1, i. e., bits from blocks $1, \dots, \kappa(i) - 1$. Note that event $A_{i,\ell}$ might imply that at least one of these bits flips to 1 for sure to maintain feasibility of the search point. We still pessimistically assume this to happen and denote by j^* the

random index of the zero-bit that is forced to flip. Furthermore, we pessimistically assume that bits $1, \dots, B - 1$ are all 1 in $x^{(t)}$ so that the contribution of bit j^* becomes as large as possible. Then, since the flips in $R(i)$ are not part of the fourth item in the definition of $A_{i,\ell}$ (Definition 5.5), we conclude that j^* is uniform on $\{B, \dots, s(\kappa(i)) - 1\}$ and contributes at most

$$\frac{1}{s(\kappa(i)) - B} \sum_{k=B}^{s(\kappa(i))-1} g_k = \frac{1}{s(\kappa(i)) - B} \sum_{k=B+1}^{s(\kappa(i))-1} g_k.$$

With respect to the bits different from j^* , we exploit that they are flipped independently. Hence, on $A_{i,\ell}$, the probability that $k \in \mathbb{Z} \cap \{B + 1, \dots, s(\kappa(i)) - 1\} \setminus \{j^*\}$ flips is bounded from above by $\frac{1}{n}$. Pessimistically, we assume that A occurs in such a mutation. By using linearity of expectation and combining with the contribution of j^* , it follows that

$$E(\Delta_R(i) | A_{i,\ell}) \leq \sum_{k=B+1}^{s(\kappa(i))-1} \frac{1}{n} g_k + \frac{1}{s(\kappa(i)) - B} \sum_{k=B+1}^{s(\kappa(i))-1} g_k$$

which is at most

$$\frac{2}{s(\kappa(i)) - B} \sum_{k=B+1}^{s(\kappa(i))-1} g_k \leq \frac{2}{s(\kappa(i)) - B} \sum_{k=B+1}^{s(\kappa(i))-1} \gamma_k,$$

where we used that $g_k \leq \gamma_k$ for all $k \in [n]$ by Definition 5.3. Along with (8) and (9), we get

$$E(\Delta_t | A_{i,\ell}) \geq \frac{g_{s(i)}}{e} - \frac{2}{s(\kappa(i)) - B} \sum_{k=B+1}^{s(\kappa(i))-1} \gamma_k$$

We are left with the sum over k . Plugging in the definition of γ_k , this is estimated by

$$\begin{aligned} \sum_{k=B+1}^{s(\kappa(i))-1} \gamma_k &= \sum_{k=1}^{s(\kappa(i))-1-B} k^7 \\ &\leq \frac{(s(\kappa(i))-B)^8}{8} \\ &\leq \frac{(s(\kappa(i))-B)g_{s(\kappa(i))}}{8} \\ &\leq \frac{(s(\kappa(i))-B)g_{s(i)}}{8}, \end{aligned}$$

where we used that $g_{s(\kappa(i))} = \gamma_{s(\kappa(i))} = (s(\kappa(i)) - B)^7$ according to the definition of $\kappa(i)$ as well as $g_{s(i)} \geq g_{s(\kappa(i))}$.

Hence, finally,

$$E(\Delta_t | A_{i,\ell}) \geq \frac{g_{s(i)}}{e} - \frac{2g_{s(i)}(s(\kappa(i)) - B)}{8(s(\kappa(i)) - B)} \geq 0.11g_{s(i)}.$$

□

5.3 Minimisation versus Maximisation

The results we presented in this paper have been formulated with respect to the minimisation of the linear function under a uniform lower constraint $x_1 + \dots + x_n \geq B$. This perspective of minimisation fits more naturally the minimisation of potential functions used in drift theorems (Theorems 2.1 and 2.2) and is therefore de-facto standard in many recent papers dealing with the optimisation of linear functions [4, 23].

However, previous works about the optimisation of linear functions under constraints considered the maximisation of a linear function under an upper uniform constraint $x_1 + \dots + x_n \leq B$. It is not difficult to see that our main theorems (Theorems 4.2 and 5.1) also hold for this scenario. Since this is rather straightforward to realise for RLS, we only discuss the result for (1+1) EA now. The potential function g from Definition 5.3 would have to be adapted to assign weight 0 to the B most significant positions and increasing weights from bits 1 to $n - B$ in the same way as before, with the exception that 0-bits instead of 1-bits would contribute: roughly speaking we would define $g(x) := \sum_{i=1}^{n-B} g_i(1-x_i)$. The log B -factor that the bound from Theorem 5.1 includes stems from the fact that once having reached a tight search point, B one-bits have to be brought to the correct positions, more precisely the B least significant ones in the case of minimisation and the B most significant ones in the case of maximisation. Hence, the log B factor will be maintained also in the scenario of maximisation and does not turn into $\log(n - B)$ as one might think at first glance.

Interestingly, the time to reach the feasible region analysed in Lemma 4.1 will be $O(n \log(n/B))$ instead of $O(n \log(n/(n - B))$ in the scenario of maximisation, as proved in earlier work [10]. This is due to the fact that a large B corresponds to a large infeasible region in the maximisation case but a small one in the minimisation case. However, in both cases the time to reach the feasible region is always bounded by an asymptotically smaller expression than our bound for the time to find an optimal search point after having reached the feasible region.

6 CONCLUSION

We have carried out a rigorous theoretical analysis on the expected optimisation time of RLS and (1+1) EA on the problem of minimising a linear function under uniform constraint. Our results include a tight expected bound of $O(n^2)$ for RLS, as well as an improved expected bound of $O(n^2 \log B)$ for (1+1) EA, where B is the constraint value, i.e., the minimum number of 1-bits that a solution should have to be considered feasible. We have also proved an upper bound of $O(n^2 \log n)$ for (1+1) EA with high probability. In order to prove our results for (1+1) EA, we have conducted an adaptive drift analysis with a potential function that depends on the weights of the linear function and the constraint value B . We are optimistic that the developed techniques can be helpful in finding upper bounds on the expected optimisation time of (1+1) EA on more complicated problems for which currently best upper bounds depend on the weights of the given input. This includes the minimum spanning tree problem where the best proven upper bound for general graphs is $O(n^2(\log n + \log w_{\max}))$ and conjectured to be $O(n^2 \log n)$.

ACKNOWLEDGEMENT

This research has been supported by the Australian Research Council (ARC) through grant DP160102401.

REFERENCES

- [1] Anne Auger and Benjamin Doerr. 2011. *Theory of Randomized Search Heuristics – Foundations and Recent Developments*. World Scientific Publishing.
- [2] Benjamin Doerr and Leslie Ann Goldberg. 2010. Adaptive Drift Analysis. In *Proc. of PPSN '10 (LNCS)*, Vol. 6238. Springer, 32–41.
- [3] Benjamin Doerr and Leslie Ann Goldberg. 2013. Adaptive Drift Analysis. *Algorithmica* 65, 1 (2013), 224–250.

- [4] Benjamin Doerr, Daniel Johannsen, and Carola Winzen. 2010. Drift Analysis and Linear Functions Revisited. In *Proc. of CEC '10*. IEEE Press, 1–8.
- [5] Benjamin Doerr, Daniel Johannsen, and Carola Winzen. 2010. Multiplicative Drift Analysis. In *Proc. of GECCO '10*. ACM Press, 1449–1456.
- [6] Benjamin Doerr, Daniel Johannsen, and Carola Winzen. 2012. Multiplicative drift analysis. *Algorithmica* 64, 4 (2012), 673–697.
- [7] Benjamin Doerr and Sebastian Pohl. 2012. Run-time analysis of the (1+1) evolutionary algorithm optimizing linear functions over a finite alphabet. In *Proc. of GECCO '12*. ACM Press, 1317–1324.
- [8] Benjamin Doerr, Dirk Sudholt, and Carsten Witt. 2013. When do evolutionary algorithms optimize separable functions in parallel?. In *Proc. of FOGA '13*. ACM Press, 51–64.
- [9] Stefan Droste, Thomas Jansen, and Ingo Wegener. 2002. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science* 276 (2002), 51–81.
- [10] Tobias Friedrich, Timo Kötzing, J.A. Gregor Lagodzinski, Frank Neumann, and Martin Schirneck. 2018. Analysis of the (1+1) EA on subclasses of linear functions under uniform and linear constraints. *Theoretical Computer Science* (2018). In press, <https://doi.org/10.1016/j.tcs.2018.04.051>.
- [11] Bruce Hajek. 1982. Hitting-Time and Occupation-Time Bounds Implied by Drift Analysis with Applications. *Advances in Applied Probability* 13, 3 (1982), 502–525.
- [12] Jun He and Xin Yao. 2004. A study of drift analysis for estimating computation time of evolutionary algorithms. *Natural Computing* 3, 1 (2004), 21–35.
- [13] Jens Jägersküpper. 2008. A Blend of Markov-Chain and Drift Analysis. In *Proc. of PPSN '08 (LNCS)*, Vol. 5199. Springer, 41–51.
- [14] Jens Jägersküpper. 2011. Combining Markov-Chain Analysis and Drift Analysis. *Algorithmica* 59, 3 (2011), 409–424.
- [15] Thomas Jansen. 2013. *Analyzing Evolutionary Algorithms - The Computer Science Perspective*. Springer.
- [16] Daniel Johannsen. 2010. *Random Combinatorial Structures and Randomized Search Heuristics*. Ph.D. Dissertation. Saarland University.
- [17] Per Kristian Lehre and Carsten Witt. 2014. Concentrated Hitting Times of Randomized Search Heuristics with Variable Drift. In *Proc. of ISAAC '14 (LNCS)*, Vol. 8889. Springer, 686–697. Extended technical report at <http://arxiv.org/abs/1307.2559>.
- [18] Boris Mitavskiy, Jonathan E. Rowe, and Chris Cannings. 2009. Theoretical analysis of local search strategies to optimize network communication subject to preserving the total number of links. *International Journal of Intelligent Computing and Cybernetics* 2, 2 (2009), 243–284.
- [19] Frank Neumann and Ingo Wegener. 2007. Randomized local search, evolutionary algorithms, and the minimum spanning tree problem. *Theoretical Computer Science* 378, 1 (2007), 32–40.
- [20] Joachim Reichel and Martin Skutella. 2009. On the size of weights in randomized search heuristics. In *Proc. of FOGA '09*. ACM Press, 21–28.
- [21] Jonathan E. Rowe and Dirk Sudholt. 2012. The Choice of the Offspring Population Size in the (1,λ) EA. In *Proc. of GECCO '12*. ACM, 1349–1356.
- [22] Ingo Wegener. 2001. Methods for the Analysis of Evolutionary Algorithms on Pseudo-Boolean Functions. In *Evolutionary Optimization*, Ruhul Sarker, Masoud Mohammadian, and Xin Yao (Eds.). Kluwer Academic Publishers.
- [23] Carsten Witt. 2013. Tight Bounds on the Optimization Time of a Randomized Search Heuristic on Linear Functions. *Combinatorics, Probability and Computing* 22, 2 (2013), 294–318.
- [24] Carsten Witt. 2014. Revised analysis of the (1+1) EA for the minimum spanning tree problem. In *Proc. of GECCO '14*. ACM Press, 509–516.