# UNIVERSITY OF DORTMUND

REIHE COMPUTATIONAL INTELLIGENCE

COLLABORATIVE RESEARCH CENTER 531

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


# Evolutionary Algorithms - How to Cope With Plateaus of Constant Fitness and When to Reject Strings of the Same Fitness


Thomas Jansen and Ingo Wegener



No. CI-96/00

Technical Report ISSN 1433-3325 September 2000

Secretary of the SFB 531 · University of Dortmund · Dept. of Computer Science/XI
44221 Dortmund · Germany

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

# Evolutionary Algorithms — How to Cope With Plateaus of Constant Fitness and When to Reject Strings of the Same Fitness*

**Thomas Jansen and Ingo Wegener**
FB Informatik, LS 2, Univ. Dortmund, 44221 Dortmund, Germany
{jansen, wegener}@ls2.cs.uni-dortmund.de

**Abstract**

The most simple evolutionary algorithm, the so-called (1+1)EA accepts a child if its fitness is at least as large (in the case of maximization) as the fitness of its parent. The variant $(1+1)^*$EA only accepts a child if its fitness is strictly larger than the fitness of its parent. Here two functions related to the class of long path functions are presented such that the $(1+1)$EA maximizes one of it in polynomial time and needs exponential time for the other while the $(1+1)^*$EA has the opposite behavior. These results prove that small changes of an evolutionary algorithm may change its behavior significantly. Since the $(1+1)$EA and the $(1+1)^*$EA differ only on plateaus of constant fitness, the results also show how evolutionary algorithms behave on such plateaus. The $(1+1)$EA can pass a path of constant fitness and polynomial length in polynomial time. Finally, for these functions it is shown that local performance measures like the quality gain and the progress rate do not describe the global behavior of evolutionary algorithms.

## 1 Introduction

Evolution strategies (Rechenberg (1994), Schwefel (1995)) are randomized search heuristics for the optimization of functions. Here we consider the maximization of pseudo-Boolean functions $f\colon \{0,1\}^n \to \mathbb{R}_0^+$. Note, that we consider a discrete search space, $\{0,1\}^n$, which substantially changes the analysis compared to the continuous space $\mathbb{R}^n$. Since this differs somehow from common analyses of evolution strategies we adopt the broader term "evolutionary algorithm". The perhaps most simple evolutionary algorithm is the so-called $(1+1)$EA with the standard mutation probability $1/n$.

**Algorithm 1 ((1 + 1)EA).**

1. *Choose randomly an initial bit string* $x \in \{0,1\}^n$.
2. *Repeat the following mutation step:*
   *Compute* $x'$ *by flipping independently each bit* $x_i$ *with probability* $1/n$.
   *Replace* $x$ *by* $x'$ *iff* $f(x') \geq f(x)$.

*This work was supported by the Deutsche Forschungsgemeinschaft (DFG) as part of the Collaborative Research Center "Computational Intelligence" (SFB 531).

In applications, we need a stopping criterion. Here the (1+1)EA is analyzed as infinite stochastic process and we are interested in two global performance measures. Let $X_f$ be the random variable describing the first point of time where $x$ is a bit string of maximal fitness. Then we are interested in the expected runtime $\mathrm{E}(X_f)$, the mean of $X_f$, and the success probability $s_f(t) = \mathrm{Prob}(X_f \leq t)$. If the success probability for a polynomial time bound $t(n)$ is not too small ($1/p(n)$ for a polynomial $p$ is enough), a multi-start strategy has a polynomial expected runtime.

Our analysis concentrates on the (1+1)EA, since more general evolutionary algorithms will have no advantage on the functions $f$ considered here. We will argue in later sections why $(\mu+\lambda)$- and $(\mu,\lambda)$-evolution strategies and even crossover will not help. Hence, our results are not only valid for the $(1+1)$EA. If the $(1+1)$EA reaches a plateau of constant fitness values, it accepts each bit string on this plateau until it finds a better one. During this time the (1+1)EA performs a kind of random walk on the plateau. It is an important problem to determine which plateaus are easy for the $(1+1)$EA. Furthermore, one may ask what we can loose if we do not perform the random walk on the plateau, i.e., if we do not accept bit strings with the same fitness. This leads to the algorithm $(1+1)^*$EA:

**Algorithm 2.** $(1+1)^*$*EA.*
*This algorithm works in the same way as the* (1+1)*EA with the exception that the condition* $f(x') \geq f(x)$ *is replaced by* $f(x') > f(x)$.

Let $X_f^*$, $\mathrm{E}\left(X_f^*\right)$, and $s_f^*(t)$ be the random variables and performance measures for the $(1+1)^*$EA which correspond to $X_f, \mathrm{E}(X_f)$, and $s_f(t)$, resp., for the $(1+1)$EA.

It is an obvious conjecture that the $(1+1)^*$EA should perform worse than the (1+1)EA. With respect to the NFL theorem (Wolpert and Macready (1997)) one has to be careful with such statements. Droste, Jansen, and Wegener (1999) have shown that there is no NFL theorem if the considered class of fitness functions is restricted to those functions which may occur in black box optimization, i. e., functions which can be evaluated in polynomial time and have a short description. Droste, Jansen, and Wegener (1998b) have considered the so-called needle-in-the-haystack function ($f(x) = 0$ if $x \neq 1^n$, $f(1^n) = 1$). Together with an improved complexity analysis by Garnier, Kallel, and Schoenauer (1999) we can conclude that $\mathrm{E}(X_f) = \Theta(2^n)$ while $\mathrm{E}\left(X_f^*\right) = \Theta(n^n)$. However, both algorithms are inefficient for this function. In Section 2 we present a function related to the long path functions due to Horn, Goldberg, and Deb (1994) and called $\mathrm{SPC}_n$ (short path with constant values on the short path) such that $\mathrm{E}(X_{\mathrm{SPC}_n})$ is polynomially bounded while $\mathrm{E}\left(X_{\mathrm{SPC}_n}^*\right)$ grows exponentially. Moreover, even multi-start variants of the $(1+1)^*$EA need exponential time on $\mathrm{SPC}_n$. The existence of such a function is not surprising. However, the analysis of this function has interesting features. In particular, we show how the $(1+1)$EA is able to reach the end of a path of polynomial length without using shortcuts and without getting hints from the fitness function. This result belongs to the many investigations of different fitness landscapes (for an overview see Bäck, Fogel, and Michalewicz (1997)).

The function $\mathrm{SPC}_n$ is also used in Section 3 to compare local performance measures like the quality gain and the progress rate with global performance measures. Obviously, we are mostly interested in the global performance. Many authors (see Bäck, Fogel, and

Michalewicz (1997)) have claimed that the microscopic view described by the local performance measures determines the macroscopic view described by the global performance measures. Such statements are true in general only if the local performance measures describe a sufficient statistics for the Markoff process. In Section 3 it is shown that most of the considered local performance measures give wrong hints for $\mathrm{SPC}_n$. The quality gain cannot find differences between the $(1+1)$EA and the $(1+1)^*$EA and the progress rate indicates that the function $\mathrm{SPC}_n$ cannot be maximized efficiently by the $(1+1)$EA.

One may ask whether the consideration of the $(1+1)^*$EA is useful at all. Menke (1998) has found a particular function (not a sequence of functions) where the $(1+1)^*$EA has a smaller expected runtime than the $(1+1)$EA. Here we present a sequence of functions denoted by $\mathrm{SPT}_n$ (short path with a trap) such that the $(1+1)$EA running for exponentially many steps has a very small success probability while the $(1+1)^*$EA running for $O(n^3 \log n)$ steps has a very high success probability. However, if we perform $cn^{5/2}$, $c$ a constant large enough, independent runs of the $(1+1)$EA, we have a good chance to find the optimum in polynomial time. Therefore, we present a more complicated function with many traps where the $(1+1)^*$EA is still successful while even multi-start variants of the $(1+1)$EA have only very small success probability.

# 2 Evolutionary Algorithms on Short Path Functions With a Constant Fitness on the Path

A path in $\{0,1\}^n$ is a sequence $x_0, x_1, \ldots, x_p$ of bit strings such that the Hamming distance between adjacent strings equals 1, i. e., $H(x_i, x_{i+1}) = 1$ where $H(x,y) = \#\{j | x_j \neq y_j\}$. A path is called *short* if it has polynomial length and *long* if it has exponential length. Horn, Goldberg, and Deb (1994) have defined long path functions where the fitness is increasing along the path. Moreover, each bit string outside the path has a smaller fitness than all the bit strings on the path and their fitness is defined in such a way that all hints lead to the beginning of the path (Rudolph (1997)). Horn, Goldberg and Deb (1994) conjectured that evolution strategies have difficulties with long path functions. Rudolph (1997) has shown that this is not the case in general, since especially for the long path functions presented by Horn, Goldberg, and Deb (1994) it is very likely that the $(1+1)$EA uses shortcuts (in this case jumps of length 3) and this leads to a polynomial expected runtime. Rudolph (1997) also has suggested long paths where shortcuts seem to be unlikely. This has been proved by Droste, Jansen, and Wegener (1998b). The $(1+1)$EA needs exponential time on these long path functions.

We start our discussion with the consideration of a short path function $\mathrm{SPI}_n$ (short path with increasing values on the path). This result will be used in Section 3 and Section 4.

**Definition 3.** *$SPI_n\colon \{0,1\}^n \to \mathbb{R}_0^+$ is defined by*

$$SPI_n(a) := \begin{cases} 2n & \text{if } a = 1^n, \\ n + \varepsilon_i & \text{if } a = 1^i0^{n-i}, 0 \le i \le n-1, \\ n - ONEMAX_n(a) & \text{otherwise,} \end{cases}$$

*where $0 < \varepsilon_0 < \varepsilon_1 < \cdots < \varepsilon_{n-1} < n$ and $ONEMAX_n(a) = a_1 + \cdots + a_n$ is the function counting the number of ones of $a$.*

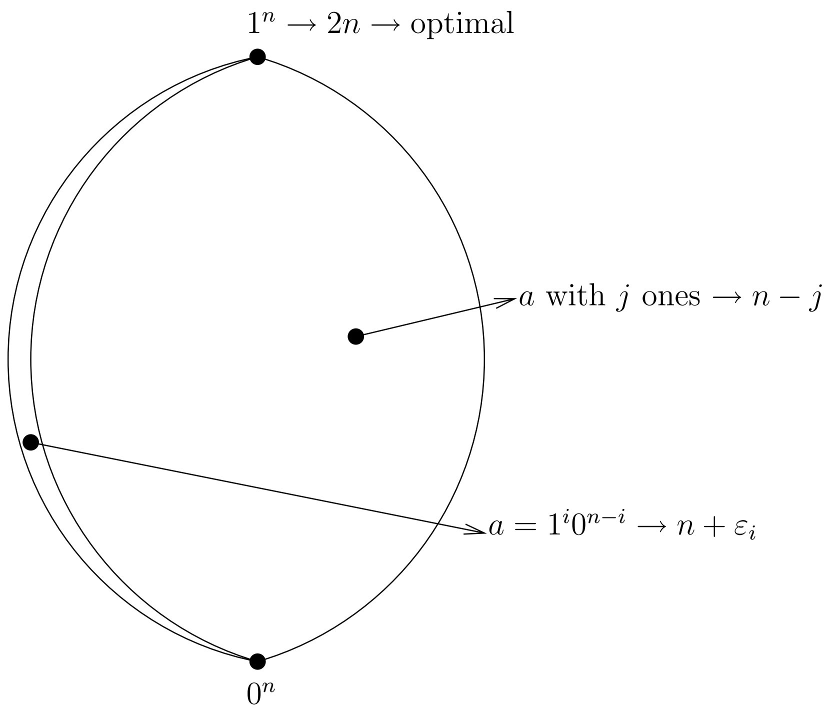



Figure 1: Illustration of the function $\mathrm{SPI}_n$, if $0 < \varepsilon_i < \varepsilon_{i+1} < n$, and of the function $\mathrm{SPC}_n$, if $\varepsilon_i = 0$ for all $i$.

**Proposition 4.** *The expected runtime of the* (1+1)*EA and the* (1+1)$^*$*EA on $SPI_n$ equals $\Theta(n^2)$.*

*Proof.* The function $\mathrm{ONEMAX}_n$ has been analyzed by many authors. The expected runtime of the $(1+1)$EA equals $\Theta(n \log n)$ (see Mühlenbein (1992) for an early proof of the upper bound and Droste, Jansen, and Wegener (1998a) for the lower bound). Both proofs also work for the $(1+1)^*$EA. Hence, the expected time to reach the path $x_0 = 0^n, x_i = 1^i0^{n-i}, 1 \le i \le n-1, x_n = 1^n$ is $O(n \log n)$. The expected time to reach some $x \in \{x_{i+1}, \ldots, x_n\}$ from $x_i$ equals $O(n)$, since the probability to obtain $x_{i+1}$ by mutation equals $1/n(1-1/n)^{n-1} \ge 1/(e \cdot n)$. Since $n$ of these steps are sufficient, the upper bound is proved.

The lower bound can be proved as follows. By Chernoff's bound (see Hagerup and Rüb (1989) or Motwani and Raghavan (1995)), the probability that the initial string has at most $(1/2+\varepsilon)n$ ones, $\varepsilon > 0$ an arbitrary constant, is exponentially close to 1, i. e., is $1-e^{-\Omega(n)}$. The probability that $\varepsilon n$ bits flip simultaneously is exponentially small. This also holds for one step among at most $n^2$ steps. Hence, with a probability exponentially close to 1, the algorithm either takes at least $n^2$ steps or reaches the path at some bit string $x_i$, $i \le (1/2+2\varepsilon)n$. (With a refined analysis we can bound $i$ even by $n^{\varepsilon}$.) The probability of a jump from $x_i$ to $x_{i+j}$ equals $(1/n)^j(1-1/n)^{n-j} = \Theta\left((1/n)^j\right)$. The probability of a jump of the length $j=3$ among $n^2$ steps is bounded by $O(1/n)$. Using only steps of length 1 and 2, the expected runtime to perform $(1/2-2\varepsilon)n$ steps on the path is $\Omega\left(n^2\right)$. □

Proposition 4 shows that evolutionary algorithms are very efficient on short paths with increasing fitness values on the path. We quickly find the path and then can follow the hints given by the fitness function. Larger populations do not help for $\text{SPI}_n$ and the crossover of two bit strings $x$ and $y$ belonging to the path, $x = 1^i0^{n-i}, y = 1^j0^{n-j}, i \le j$, can only create bit strings $z$ outside the path or "between" $x$ and $y$, i. e., $z = 1^k0^{n-k}$ where $i \le k \le j$.

**Definition 5.** *$SPC_n\colon \{0,1\}^n \to \mathbb{R}_0^+$ (short path with constant values on the short path) is defined by*

$$SPC_n(a) := \begin{cases} 2n & \text{if } a = 1^n, \\ n & \text{if } a = 1^i0^{n-i}, 0 \le i \le n-1, \\ n - ONEMAX_n(a) & \text{otherwise.} \end{cases}$$

The function $\text{SPC}_n$ differs from $\text{SPI}_n$ in one essential aspect. Only the terminal string on the path is better than the rest of the path. The path from $0^n$ to $1^{n-1}0$ is a plateau of bit strings with the same fitness value. Is it possible for evolutionary algorithms to find the "golden terminal" of the path in polynomial time? Our results easily can be generalized to shorter and longer paths as long as their length is polynomial. First, we show that the $(1+1)^*$EA is inefficient for $\text{SPC}_n$ implying that in this situation a random exploration of the plateau is necessary.

**Theorem 6.** *The success probability of the $(1+1)^*EA$ on $SPC_n$ within $n^{n/2}$ steps is exponentially small, i. e., bounded by $e^{-\Omega(n)}$. The expected runtime is bounded below by $n^{\Omega(n)}$.*

*Proof.* The result on the expected runtime follows from the result on the success probability. The function $\text{SPC}_n$ takes only $n+1$ different values implying that the $(1+1)^*$EA accepts the new bit string at most $n$ times. We investigate a typical run of the $(1+1)^*$EA and estimate the so-called "failure probability" of each phase, i. e., the probability that the run of the $(1+1)^*$EA is not typical. We prove that a typical run takes $n^{n/2}$ steps and that the sum of all failure probabilities is bounded by $e^{-\Omega(n)}$.

A typical run starts with an initial string with at most $(2/3)n$ ones. The bound on the failure probability follows from Chernoff's bound.

A typical run meets the path within $O(n^2 \log n)$ steps. The expected time to reach the path is $O(n \log n)$, since the function behaves like $-\text{ONEMAX}_n$ outside the path. Hence, by Markoff's inequality, the success probability for $O(n \log n)$ steps is at least $1/2$. This holds independently from the initial string. Within $\Omega(n^2 \log n)$ steps we have $n$ phases of length $\Theta(n \log n)$. Hence, the failure probability is bounded by $e^{-\Omega(n)}$.

A typical run meets the short path at some string $x = 1^i 0^{n-i}$ where $i \leq n/3$. Here we can assume that the failures considered above did not happen. We investigate the $(1+1)^*$EA on $-\text{ONEMAX}_n$. It accepts at most one string on the $i$-th level, i. e., one string with $i$ ones. This is a random string among all strings on the $i$-th level. For all levels $l$, $n/3 \leq l \leq 2n/3$, the probability to find a string $x$ whose Hamming distance to $1^l 0^{n-l}$ is smaller than $n/10$ is $e^{-\Omega(n)}$. Hence, also the probability that a mutation step leads to a string on the path is $e^{-\Omega(n)}$. This holds for the whole phase of length $\Theta(n^2 \log n)$ until we reach the path.

If all considered failures do not happen, the path is met at some string $1^i 0^{n-i}$ where $i \leq n/3$. In order to reach $x_{\text{opt}} = 1^n$ it is necessary that all zeros flip simultaneously and all ones do not flip. This probability is bounded above by $(1/n)^{2n/3}$. The probability that such an event happens within $n^{n/2}$ steps is bounded above by $e^{-\Omega(n)}$. Altogether, we have proved the theorem. □

**Theorem 7.** *The expected runtime of the* $(1+1)$*EA on SPC*$_n$ *is bounded above by* $O(n^3)$*. The success probability for* $n^4$ *steps is bounded below by* $1 - e^{-\Omega(n)}$.

*Proof.* We prove that the success probability for $cn^3$ steps, where $c$ is some large enough constant, is bounded below by some constant $\alpha > 0$. This implies both statements, since the statement holds for arbitrary initial strings. In the same way as in the proof of Theorem 6 we can assume that the path is reached within $O(n^2 \log n)$ steps. The failure probability is $e^{-\Omega(n)}$.

In the following we have to investigate the $(1+1)$EA on the path from $x_0 = 0^n$ via $x_i = 1^i 0^{n-i}$, $1 \leq i \leq n-1$, to $x_n = 1^n$. The $(1+1)$EA accepts each string on the path until $1^n$ is reached. There are unsuccessful steps where $x$ mutates to some $x'$ outside the path. A step is called successful if it produces a string $x'$ which differs from the current string $x$ and which is accepted as new current string. The probability of a successful step is at least $1/(en)$, since each point on the path has a Hamming neighbor on the path. For each $c' > 0$ there exists some $c$ such that the probability of less than $c'n^2$ successful steps among $cn^3$ steps is $e^{-\Omega(n)}$. This again follows from Chernoff's bound.

Now we investigate the $c'n^2$ successful steps where $c'$ can be chosen large enough. The conditional probability that a successful step mutates $k$ bits equals $\Theta\left((1/n)^{k-1}\right)$ for all $k$ and the probability that a successful step mutates at least 4 bits equals $\Theta\left((1/n)^3\right)$. A typical run of $c'n^2$ successful steps consists of

– no successful step flipping at least 4 bits (failure probability $O(1/n)$ by direct calculation),

– some successful steps flipping 3 bits,

– some successful steps flipping 2 bits,

– and at least $c''n^2$ successful steps flipping exactly one bit where $c''$ depends on $c'$ and can be chosen arbitrarily large if one increases $c'$ (the conditional probability that a successful step flips more than one bit is bounded by $O(1/n)$, this implies by Chernoff's bound that the number of successful steps flipping more than one bit is with large probability bounded above by $n^{3/2}$, the failure probability is $e^{-\Omega(n)}$).

We claim that the probability to reach one of the last four strings of the path, namely $1^{n-3}0^3, 1^{n-2}0^2, 1^{n-1}0, 1^n$, is bounded below by a positive constant. If we do not have reached one of the target strings, the probability that a successful step of length $l \in \{1, 2, 3\}$ increases the number of ones is at least $1/2$. With probability at least $1/2$ at least half of the successful steps of length 3 increase the number of ones of the current string. The same holds for the successful steps of length 2 (independently of the steps of length 3). Finally, we consider the successful steps of length 1. Here we use another notion of "success". A step is a success if $x'$ is closer to $1^n$ than $x$. We have $c''n^2$ independent Bernoulli trials with success probability $1/2$. We want to estimate the probability of at least $(1/2)c''n^2 + (1/2)n$ successes. The probability of less than $(1/2)c''n^2$ successes is bounded above by $1/2$. The probability of exactly $k$ successes is bounded above by $\binom{N}{N/2}2^{-N}$ for $N = c''n^2$. By Stirling's formula, this probability can be bounded by $b \cdot N^{-1/2}$ for some constant $b$. By choosing $c''$ large enough, this probability can be bounded by $1/(2n)$. This implies that the probability of less than $(1/2)c''N^2 + (1/2)n$ successes is bounded above by $(1/2)+(n/2)\cdot(1/(2n)) = 3/4$ and the probability of at least $(1/2)c''n^2 + (1/2)n$ successes is bounded below by $1/4$. Altogether, the probability that the number of ones increases by $c'n^2$ successful steps by at least $n$ is at least $1/16 - o(1)$ taking into account all possible failures. Hence, we reach one of the last four strings of the path with probability $1/16 - o(1)$. Having reached $1^{n-j}0^j, 1 \le j \le 3$, there is a probability of $(1/2)^j - o(1)$ that each of the next $j$ successful steps increases the number of ones by one and we reach $1^n$. Altogether the probability to reach $1^n$ within $cn^3$ steps, $c$ a constant large enough, is bounded below by $(1/128) - o(1)$ and this proves the theorem. □

# 3 Local Vs. Global Performance Measures

Local performance measures consider the behavior of evolutionary algorithms within one step or at most a small number of steps.

**Definition 8.** *i) The quality gain $Q_f^A(a)$ of an evolutionary algorithm $A$ with population size 1 is the expected value of $f(x) - f(a)$ where $x$ is the string describing the random string created in one step from the string $a$.*

*ii) The progress rate $r_f^A(a)$ of an evolutionary algorithm $A$ with population size 1 working on $\{0,1\}^n$ and a function $f$ with a unique optimum $x_{opt}$ is the expected value of $H(a, x_{opt}) - H(x, x_{opt})$ ($H$ is the Hamming distance) where $x$ is the string describing the random string created in one step from the string $a$.*

The following simple result follows from the definition.

**Proposition 9.** *The local performance measure quality gain cannot distinguish between the* $(1+1)$*EA and the* $(1+1)^*$*EA.*

This trivial fact implies that the quality gain would predict for $\mathrm{SPC}_n$ the same behavior for the $(1+1)$EA and the $(1+1)^*$EA, although their global performances differ a lot: the $(1+1)^*$EA is very inefficient while the $(1+1)$EA is efficient. Moreover, the quality gain shows only very small differences between $\mathrm{SPC}_n$ and $\mathrm{SPI}_n$ (for small $\varepsilon_i$). However, the $(1+1)^*$EA behaves quite differently on these fitness functions. The quality gain for $\mathrm{SPC}_n$ can be estimated as follows:

– $Q^{(1+1)}_{\mathrm{SPC}_n}(a) = Q^{(1+1)^*}_{\mathrm{SPC}_n}(a) = 0$ for $a = 1^n$ (this always holds for optimal strings),

– $Q^{(1+1)}_{\mathrm{SPC}_n}(a) = Q^{(1+1)^*}_{\mathrm{SPC}_n}(a) = \Theta\left((1/n)^{n-i-1}\right)$ for $a = 1^i0^{n-i}$, $0 \le i \le n-1$,

– $Q^{(1+1)}_{\mathrm{SPC}_n}(a) = Q^{(1+1)^*}_{\mathrm{SPC}_n}(a) \ge i/(e \cdot n)$ for all other $a$ with $i$ ones.

The quality gain is quite large outside the path but exponentially small on the path. A small part of the search space with a very small quality gain is an obstacle only if this part is reached with large probability. No local performance measure makes a forecast whether this small part will be reached. We know that the short path is reached with high probability. The quality gain on the path is very small and with respect to the quality gain one would not expect that the $(1+1)$EA is efficient for $\mathrm{SPC}_n$.

The consideration of the progress rate for $\mathrm{SPC}_n$ is interesting. For all strings $a$ outside the path the progress rate is the same for the $(1+1)$EA and the $(1+1)^*$EA. For strings on the path, the $(1+1)$EA accepts all strings on the path. The progress rate of the $(1+1)$EA and the $(1+1)^*$EA for $\mathrm{SPC}_n$ can be estimated as follows:

– $r^{(1+1)}_{\mathrm{SPC}_n}(a) = r^{(1+1)^*}_{\mathrm{SPC}_n}(a) = 0$ for $a = 1^n$,

– $r^{(1+1)^*}_{\mathrm{SPC}_n}(a) = +\Theta\left((1/n)^{n-i}(n-i)\right)$ for $a = 1^i0^{n-i}$, $0 \le i \le n-1$,

– $r^{(1+1)}_{\mathrm{SPC}_n}(a) = -\Theta\left((n-i+1)(1/n)^{n-i+1}\right)$ for $a = 1^i0^{n-i}, i > n/2$,

– $r^{(1+1)}_{\mathrm{SPC}_n}(a) = +\Theta\left((i+1)(1/n)^{i+1}\right)$ for $a = 1^i0^{n-i}, i < n/2$,

– $r^{(1+1)}_{\mathrm{SPC}_n}(a) = r^{(1+1)^*}_{\mathrm{SPC}_n}(a) = -\Theta(i/n)$ for all $a$ with $i$ ones and lying outside the path.

The progress rate indicates that $\mathrm{SPC}_n$ is very difficult. It is positive for the $(1+1)$EA only on the first half of the path where we are far away from the optimum. On the second half of the path we have the strings $1^i0^{n-i}$ with $i > n/2$. We reach $1^{i+j}0^{n-i-j}$, if $i + j \le n$, with the same probability as $1^{i-j}0^{n-i+j}$ and we have a positive probability of reaching the earlier strings of the path. The progress rate outside the path is negative, since the hints of $\mathrm{SPC}_n$ point to $0^n$. Although only $n/2$ strings have a positive progress rate and although these positive values are very small and although all strings with a positive progress rate are far away from the optimum, the $(1+1)$EA is efficient for $\mathrm{SPC}_n$.

Moreover, the progress rate indicates that the $(1+1)^*$EA is better than the $(1+1)$EA for $\text{SPC}_n$ which is totally wrong.

We conclude that the knowledge of the behavior of local performance measures can be useful. However, there are functions like $\text{SPC}_n$ where the local performance measures lead to wrong indications.

# 4 Two Functions Where Strings of the Same Fitness Should Be Ignored

It seems to be obvious that it is better to explore a plateau of search points with the same fitness than to wait at the point where the plateau is reached for the first time for a step to some point with a better fitness. This will be the case in most situations of fitness functions which are easy to evaluate and to describe. However, this statement is not true for all these functions. The following function has the property that the plateau is reached with high probability close to some specific point. Sitting at this point it is quite likely to find the optimum efficiently but not too quickly. If one starts an exploration of the plateau during the waiting time, it is very likely that it takes a long time to come back close to the specific point and this makes it unlikely to leave the plateau.

In order to simplify the notation we assume that $n = m^2$ for some integer $m$. The statements hold for all $n$ and $m = \lfloor n^{1/2} \rfloor$.

**Definition 10.** *The function $SPT_n \colon \{0,1\}^n \to \mathbb{R}_0^+$ (short path with a trap) is defined by*

$$SPT_n(a) = \begin{cases} 3n & \text{if } a \in A = \{a^*\},\ \text{where } a^* = 0001^{n-m-3}0^m, \\ 2n & \text{if } a \in B,\ \text{i.e., } a \text{ starts with } n-m \text{ ones}, \\ n+i & \text{if } a \in C = \{1^i0^{n-i}, 0 \le i \le n-m-1\}, \\ n - ONEMAX(a) & \text{if } a \in D = \{0,1\}^n - (A \cup B \cup C). \end{cases}$$

With high probability, the search starts in $D$ and efficiently finds the short path $C$ at one of the strings in the initial part of $C$. The search follows this path efficiently and reaches the plateau $B$ at $1^{n-m}0^m$ or close to this string. There is a 3-bit-mutation from $1^{n-m}0^m$ to the optimal string $a^*$, the only string contained in $A$. If the search explores the plateau it performs a random walk in $B$ and all but an exponentially small fraction of the strings of $B$ have a Hamming distance of at least $m/5$ to $a^*$ which requires a specific mutation with at least $m/5$ flipping bits. The waiting time for such an event is exponentially large. These ideas are made precise in the rest of this section.

We partition $B$ into the sets $B_i$ of all $b$ with $i$ ones among the last $m$ positions. Let $b^* = 1^{n-m}0^m$ be the only string in $B_0$. First we are interested in the random time $T_{A \cup B}$ until the $(1+1)$EA and the $(1+1)^*$EA enter $A \cup B$ and in the probability distribution $p$ describing where the algorithms enter $A \cup B$.

**Proposition 11.** *The following results hold for the $(1+1)$EA and the $(1+1)^*$EA and the function $SPT_n$.*

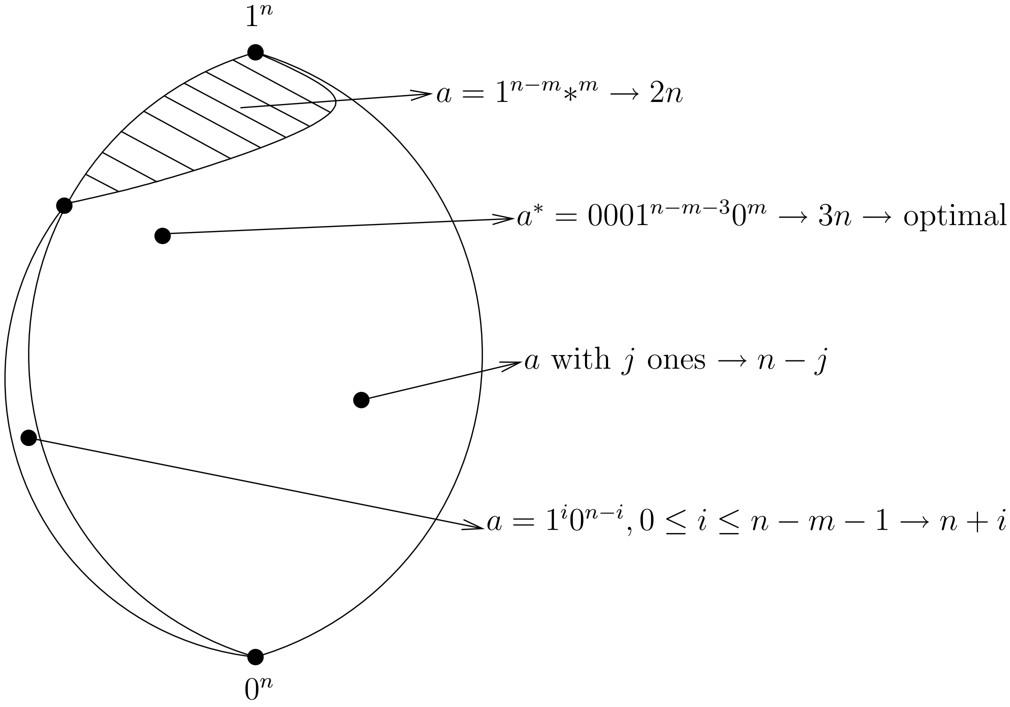



Figure 2: Illustration of the function $\text{SPT}_n$.

- $Prob\left(T_{A\cup B} = O\left(n^2 \log n\right)\right) = 1 - O(1/n)$,
- $Prob\left(T_{A\cup B} = O\left(n^3\right)\right) = 1 - e^{-\Omega(n)}$,
- $p\left(a^*\right) = \Theta\left((1/n)^3\right)$,
- $p\left(B - \{b^*\}\right) = \Theta\left((1/n)^{1/2}\right)$,
- $p\left(B_i\right) = \Theta\left(\binom{m}{i}(1/n)^i\right), 1 \leq i \leq m$, *and*
- $p\left(b^*\right) = p\left(B_0\right) = 1 - \Theta\left((1/n)^{1/2}\right)$.

*Proof.* The proof works for both algorithms, although they differ in their behavior on $D$. However, they have the same behavior on $C$. The search starts with probability $1-e^{-\Omega(n)}$ with a string with at most $(1/2+\varepsilon)n$ ones and enters $A \cup B \cup C$ at some string with at most $(1/2+2\varepsilon)n$ ones. For our purposes it is sufficient to set $\varepsilon = 1/10$. Moreover, the set $A \cup B \cup C$ is reached within $O\left(n^2 \log n\right)$ steps with a probability of $1 - e^{-\Omega(n)}$, since we have $\Omega(n)$ phases of length $O(n \log n)$ and can apply the results for $\text{ONEMAX}_n$. All strings from $A \cup B \cup C$ with at most $(1/2+2\varepsilon)n$ ones belong to $C$. (Since our results are asymptotical ones, we may assume that $n$ is large enough.)

Let $c$ be the last string reached in $C$, before the algorithm enters $A \cup B$. We denote by $mut(c \rightarrow E)$ the probability that the string obtained by mutation of $c$ belongs to some set $E$. If $c = 1^{n-m-j}0^{m+j}$,

- $mut(c \rightarrow a^*) = \Theta\left((1/n)^{j+3}\right)$,

– $mut(c \to B_i) = \Theta\left(\binom{m}{i}(1/n)^{j+i}\right), 1 \leq i \leq m$,

– $mut(c \to B - \{b^*\}) = \Theta\left(1/n)^{j+1/2}\right)$,

– $mut(c \to b^*) = \Theta\left((1/n)^j\right)$.

Independently of $j$, we obtain the same asymptotical probabilities $mut^*$ under the condition that the result of the mutation belongs to $A \cup B$, namely

– $mut^*(c \to a^*) = \Theta\left((1/n))^3\right)$,

– $mut^*(c \to B_i) = \Theta\left(\binom{m}{i}(1/n)^i\right), 1 \leq i \leq m$,

– $mut^*(c \to B - \{b^*\}) = \Theta\left((1/n)^{1/2}\right)$,

– $mut^*(c \to b^*) = 1 - \Theta\left((1/n)^{1/2}\right)$.

This implies the results on the probability distribution $p$. The results on $T_{A \cup B}$ follow from the proof of Proposition 4 that the expected time to reach the terminal of a path of linear length with increasing fitness values equals $O\left(n^2\right)$ independently from the starting point. □

**Theorem 12.** *Let $k \geq 0$ be an integer. The success probability of the $(1+1)^*$EA on $SPT_n$ within $O\left(n^{3+k}\log n\right)$ steps is bounded below by $1 - O\left((1/n)^{(k+1)/2}\right)$.*

*Proof.* By Proposition 11, we can assume that $A \cup B$ is reached within $O\left(n^3\right)$ steps. We also know the probability distribution describing where $A \cup B$ is entered. If we have reached $a^*$, the search was successful. Otherwise, the $(1+1)^*$EA has reached some $b \in B_i$ and accepts only $a^*$. Hence, we wait for a success within independent Bernoulli trials with success probability $\Theta\left((1/n)^{i+3}\right)$. The expected waiting time equals $\Theta\left(n^{i+3}\right)$. The failure probability for $\Theta\left(n^{i+3}\log n\right)$ steps equals $\Theta(1/n)$ and for $\Theta\left(n^{i+4}\log n\right)$ steps it is exponentially small.

Now we investigate the probability that the $(1+1)^*$EA has not found $a^*$ within $\Theta\left(n^{3+k}\log n\right)$ steps. This happens if $b \in B_i$ and $\Theta\left(n^{3+k}\log n\right)$ steps are not sufficient to reach $a^*$ from $b$. The probability of this event is bounded above by 1, if $i > k$, by $O(1/n)$, if $i = k$, and $e^{-\Omega(n)}$, if $i < k$. With the results of Proposition 11 describing where we enter $A \cup B$, we obtain the following upper bound on the failure probability where the three terms reflect the three cases $i > k$, $i = k$, and $i < k$ considered above (remember that $m = n^{1/2}$):

$$1 \cdot O\left(\binom{m}{k+1}\left(\frac{1}{n}\right)^{k+1}\right) + O\left(\frac{1}{n}\right) \cdot O\left(\binom{m}{k}\left(\frac{1}{n}\right)^{k}\right) + e^{-\Omega(n)} \cdot 1 = O\left(\left(\frac{1}{n}\right)^{(k+1)/2}\right).$$

For constant $k$, it is possible to bound the probability to enter $A \cup B$ in some $B_i$, $i > k$, by $O\left(p(B_{k+1})\right)$. □

Theorem 12 implies that the success probability within $O\left(n^3 \log n\right)$ steps tends to 1. Nevertheless, the expected runtime of the $(1{+}1)^*$EA grows exponentially. The probability to enter $A \cup B$ in $B_{m/2}$ equals $\Theta\left(2^m m^{-1/2}(1/n)^{m/2}\right)$ and the expected waiting time in this situation equals $\Theta\left((1/n)^{m/2+3}\right)$. In applications, a success probability tending to 1 is good enough.

In the following, we prove that the $(1+1)$EA has a very small success probability even if we allow more than polynomially many steps.

**Theorem 13.** *The success probability of the* $(1+1)$*EA on* $SPT_n$ *for* $O\left(n^2 \log n\right)$ *steps is bounded below by* $\Omega\left((1/n)^{5/2}\right)$. *The success probability for* $2^{\varepsilon n^{1/2}}$ *steps,* $\varepsilon > 0$ *small enough, is bounded above by* $O\left((1/n)^{5/2}\right)$.

*Proof.* The first result is quite easy. Proposition 11 ensures that with probability $1-o(1)$ the set $A \cup B$ is entered within $O\left(n^2 \log n\right)$ steps at the point $b^*$. The $(1+1)$EA accepts in $b^*$ all strings from $A \cup B$. The probability to reach a string $b \in B - \{b^*\}$ is $\Omega\left((1/n)^{1/2}\right)$ and the probability to reach $a^*$ equals $\Theta\left((1/n)^3\right)$. Hence, the probability to reach $a^*$ before some $b \in B - \{b^*\}$ equals $\Theta\left((1/n)^{5/2}\right)$ and the probability to leave $b$ within $O\left(n^2 \log n\right)$ steps is $1 - o(1)$.

It is more difficult to prove the upper bound on the success probability. The proof of Proposition 11 shows that the probability of reaching $A \cup B \cup C$ in $A \cup B$ or at some $c$ with more than $(7/10)n$ ones is $e^{-\Omega(n)}$. Hence, we assume in the following that we have reached the path at $c$ with at most $(7/10)n$ ones. At $c$, the probability to reach the next string on the path is $\Theta(1/n)$. If $c = 1^{n-i}0^i$, the probability to reach $a^*$ is $\Theta\left((1/n)^{i-m+3}\right)$. Hence, the probability that we reach $a^*$ if we leave $c$ is bounded above by $O\left((1/n)^{i-m+2}\right)$. Altogether, the probability to reach $a^*$ as the first string from $A \cup B$ is bounded above by $O\left((1/n)^3\right)$.

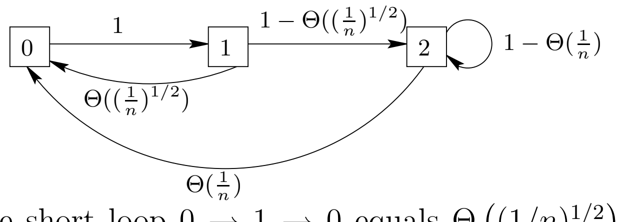
Hence, we can assume that we have reached $A \cup B$ in some string in $B$. The $(1+1)$EA has the best success probability if $A \cup B$ is reached in $b^*$. We consider $2^{\varepsilon m}$ successful steps of the $(1+1)$EA, i. e., steps where some string different from the current one is accepted. Let $x_0 = b^*$ and let $x_t$ be the current string after $t$ steps under the condition that $x_i \neq a^*$ for all $i < t$. We prove an upper bound $\alpha(t)$ on the probability to reach $a^*$ during the next successful step. If $x_t \in B_j$, $\alpha(t) = O\left((1/n)^{5/2}\right)$, since there are $m$ 1-bit-mutations leading to a successful step and, therefore, the probability of a successful step is $\Omega\left((1/n)^{1/2}\right)$ and the probability to obtain $a^*$ from $x_t \in B_j$ by mutation is $\Theta\left((1/n)^{j+3}\right)$.

As long as the $(1+1)$EA explores $B$, the current string starts with $n-m$ ones and a step is unsuccessful if one of these bits flips. Hence, we consider the current bit string as a string of length $m$ where we are starting with $0^m$. Let $y_t$ be the random string after $t$ mutations (where still each bit flips with probability $1/n$). Here we consider also steps which do not change one of the last $m$ bits. The $i$-th bit of $y_t$ equals 1 iff this bit has flipped for an odd number of times. The number of flips of each bit is binomially distributed with parameters $t$ and $1/n$. For $t = n$ this distribution can be approximated by the Poisson distribution with parameter $\lambda = 1$. Hence, the probability that the $i$-th bit of $y_t$ equals 1 converges to $e^{-1}$ and is larger than $1/4$ for $n$ large enough. The bits of $y_t$ are flipped independently. Hence, by Chernoff's bound, the number of ones in $y_t$ is at least $m/5$ with a probability of $1 - e^{-\Omega(m)}$. This also implies that the probability of reaching $a^*$ with the next successful step is $O\left((1/n)^{m/5+5/2}\right)$.

We can generalize these calculations to all $t \geq n$, since the probability that the $i$-th bit of $y_t$ equals 1 converges monotonically to $1/2$. Hence, the success probability of $2^{\varepsilon m} = 2^{\varepsilon n^{1/2}}$ steps ($\varepsilon > 0$ some constant small enough) is exponentially small if the $(1+1)$EA was not successful before reaching $B$ and during the first $n$ steps after having reached $B$.

We still have to estimate the success probability for the first $n$ steps after having reached $B$. The probability that a step is successful is $\Theta\left((1/n)^{1/2}\right)$. Hence, by Chernoff's bounds, there are with large probability at most $\Theta\left(n^{1/2}\right)$ successful steps. The probability that $a^*$ is reached by a successful step is bounded by $O\left((1/n)^{5/2}\right)$. Hence, the probability of reaching $a^*$ within $\Theta\left(n^{1/2}\right)$ successful steps is bounded by $O\left((1/n)^2\right)$. We can improve this rough estimate. All successful steps from some string $b \in B - B_0$ have a probability to reach $a^*$ which is even bounded by $O\left((1/n)^{7/2}\right)$. Hence, the probability to reach $a^*$ within $\Theta\left(n^{1/2}\right)$ successful steps starting in $B - B_0$ is bounded above by $O\left((1/n)^3\right)$. We look for a bound how often a successful step starts from $b^*$, the single string contained in $B_0$.

This estimation is performed using a simplified Markoff chain with three states 0, 1 and 2 representing $B_0$, $B_1$ and $B - (B_0 \cup B_1)$ resp. Since we consider successful steps not reaching $a^*$, the state 0 is left with probability 1 within one step. The probability to reach $b^*$ again is overestimated if we assume that each step from state 0 reaches state 1. A successful step from state 1 reaches state 0 with probability $\Theta\left((1/n)^{1/2}\right)$ (one appropriate 1-bit-mutation), state 1 with probability $\Theta(1/n)$ ($m-1$ appropriate 2-bit-mutations), and state 2 with the remaining probability of $1-\Theta\left((1/n)^{1/2}\right)$. We overestimate the probability to reach state 0 by adding the probability to stay in state 1 to the probability to reach state 0. A successful step from state 2 reaches state 0 with probability $O\left((1/n)^{3/2}\right)$ (one appropriate mutation flipping at least 2 bits), reaches state 1 with probability $O\left((1/n)^{1/2}\right)$ (the probability is maximal for strings $b \in B_2$ where we have one appropriate 1-bit-mutation) and stays in state 2 with the remaining probability of $1 - O\left((1/n)^{1/2}\right)$. We overestimate the probability to reach state 0 by assuming a probability of $\Theta(1/n)$ to reach state 0 from state 2 and a probability of 0 to reach state 1 (the probability to reach state 0 via state 1 is $O(1/n)$). This leads to the following Markoff chain with the corresponding transition probabilities:




The probability of the short loop $0 \to 1 \to 0$ equals $\Theta\left((1/n)^{1/2}\right)$. The probability of using this loop six times consecutively equals $\Theta\left((1/n)^3\right)$. The probability that this happens within $\Theta\left(n^{1/2}\right)$ steps is bounded by $\Theta\left(1/n\right)^{5/2}$. Hence, we can assume that we reach state 0 at most six times before reaching state 2. The probability to reach state 0 from state 2 equals $\Theta(1/n)$. The probability that this happens 5 times within $\Theta\left(n^{1/2}\right)$

trials is bounded by $O\left((1/n)^{5/2}\right)$. Hence, with a failure probability of $O\left((1/n)^{5/2}\right)$ we have at most 30 trials to reach $a^*$ from $b^*$ which proves the theorem. $\square$

Theorem 13 implies that the success probability within the first $O(n^2 \log n)$ steps equals $\Theta\left((1/n)^{5/2}\right)$ (because of the lower bound for $2^{\epsilon n^{1/2}}$ steps) and that the success probability increases during the following $2^{\epsilon n^{1/2}}$ steps only by a constant factor. There is only one short period of time with a good chance to find the optimum.

A multi-start strategy only has a good success probability if we work with $\Omega\left(n^{5/2}\right)$ multi-starts. Altogether, the $(1+1)^*$EA is much more efficient on $\mathrm{SPT}_n$ than the $(1+1)$EA.

In the following we present a function where the $(1+1)^*$EA has an overwhelming success probability for a small polynomial number of steps but the success probability of the $(1+1)$EA is exponentially small — even after an exponential number of steps. Hence, multi-start strategies do not help. The function is based on a sequence of traps. First, we present a function with only one of these traps. The purpose is to develop the main technical results. This function is called PT (plateau with a trap). We have a main path $M$ of length $2k+1$ where $k = \lfloor n^{1/2} \rfloor$. The middle point of this path has a very bad fitness. Besides this the fitness is increasing along the path. The point with the bad fitness is called the "hole" in the path. Just before the hole another path $P$ of length $m = \lfloor n^{1/4} \rfloor$ is starting. All but the last point on this path have the same fitness as the starting point, namely the point before the hole. The last point has the largest fitness. The set of points with the same fitness is called the plateau, the last point is called the trap. It is quite obvious that neither the $(1+1)$EA nor the $(1+1)^*$EA have difficulties to find points in the last quarter of $M$ or the trap. If the last quarter of $M$ is reached before the trap, this is called a success. Otherwise, the algorithm is called "trapped". The reason is that we later define a function where the way to the global optimum leads along $M$.

**Definition 14.** *Let $k = \lfloor n^{1/2} \rfloor$ and $m = \lfloor n^{1/4} \rfloor$. The main path consists of the points in $M = \{1^i 0^{n-i} \mid 0 \le i \le 2k, i \ne k\}$, the set $H$ only contains the hole $h^* = 1^k 0^{n-k}$. The plateau consists of the points in $P = \left\{1^{k-1} 0^{n-k+1-j} 1^j \mid 1 \le j \le m-1\right\}$, the set $T$ only contains the trap $t^* = 1^{k-1} 0^{n-k+1-m} 1^m$. The function $PT_n$ is defined by*

$$PT_n(a) = \begin{cases} -2 & \text{if } a = h^* \\ i & \text{if } a = 1^i 0^{n-i} \in M \\ k-1 & \text{if } a \in P \\ n & \text{if } a = t^* \\ -1 & \text{otherwise.} \end{cases}$$

*Let $s(n)$ resp. $s^*(n)$ be the probability that the $(1+1)$EA resp. the $(1+1)^*$EA starting at $0^n$ reaches one of the last $k$ points of $M$ before reaching $t^*$.*

**Lemma 15.** *$s^*(n) = 1 - (1/n)^{\Omega(m)}$. Moreover, if one the last $k$ points of $M$ is reached before $t^*$, this point is reached within $O\left(n^{5/2}\right)$ steps with a probability of at least $1 - (1/2)^{\Omega(m)}$.*

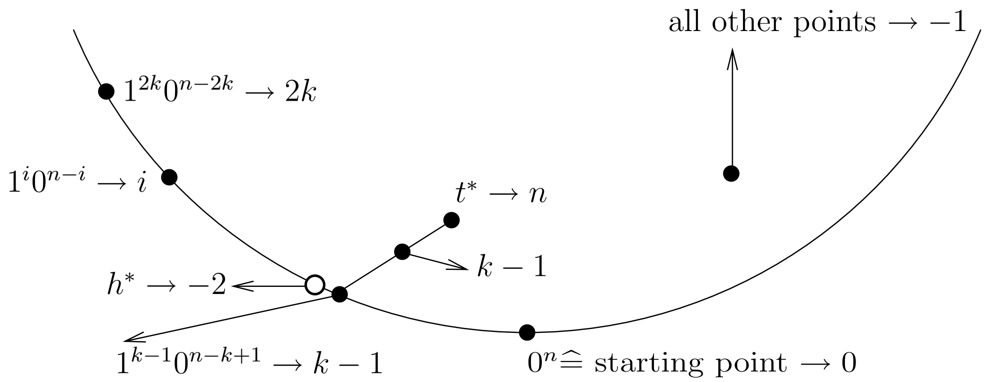



Figure 3: Illustration of the function $\mathrm{PT}_n$.

*Proof.* Let $M_1$ be the part of $M$ before $1^{k-1}0^{n-k+1}$, $P^* = \left\{1^{k-1}0^{n-k+1}\right\}$, and $M_2 = M \backslash M_1$. As long as the actual string belongs to $M_1$ all strings in $M_2 \cup P \cup P^* \cup T$ are accepted as new strings by the $(1+1)^*$EA. For each $a \in M_1$ we get the following conditional probabilities to reach the points from $M_2 \cup P \cup P^* \cup T$ (under the condition that a point in this set is reached):

$$\begin{aligned} 1^{k-1}0^{n-k+1} &: \quad \Theta(1) \\ 1^{k-1}0^{n-k+1-j}1^j &: \quad \Theta\left((1/n)^j\right) \text{ for } j \geq 1 \\ M_2 &: \quad \Theta\left((1/n)^2\right). \end{aligned}$$

Hence, with a probability of $1-(1/n)^{\Omega(m)}$ the set $M_2 \cup P \cup P^* \cup T$ is reached within $M_2 \cup P^*$ or one of the first $\lfloor m/10 \rfloor$ points of $P$. Sitting on $P \cup P^*$ no other point of this set is accepted by the $(1+1)^*$EA. The Hamming distance to the first point of $M_2$ is at most $\lfloor m/10 \rfloor - 2$ while the distance to $t^*$ is at least $(9/10)m$. Hence, with a probability of $1-(1/n)^{\Omega(m)}$ the $(1+1)^*$EA reaches at first a point from $M_2$ before reaching $t^*$.

For the second claim, we can assume that if $P \cup P^*$ is reached, it is reached within the first $\lfloor m/10 \rfloor$ points or in $P^*$. Assuming that afterwards a point in $M_2$ is reached, we obtain for the conditional expected time of this event the upper bound

$$O\left(n^2 \cdot 1 + n^3 \cdot \frac{1}{n} + n^4 \cdot \frac{1}{n^2} + \cdots + n^{\lfloor m/10 \rfloor + 2} \cdot \frac{1}{n^{\lfloor m/10 \rfloor}}\right) = O\left(n^2 \cdot m\right) = O\left(n^{9/4}\right).$$

By Markoff's inequality, there is a probability of at most 1/2 that $O\left(n^{9/4}\right)$ steps are not sufficient. The bound follows by considering $m$ phases of this length. We still have to consider the time to reach one of the points in $P \cup P^* \cup M_2$. There is a short path of length $O\left(n^{1/2}\right)$ with increasing fitness values leading to $P^*$. The expected time to reach the end of this path is $O\left(n^{3/2}\right)$ and we additionally have the chance to reach points in $P \cup M_2$. It is sufficient to consider $m$ phases of length $O\left(n^{3/2}\right)$. □

The function $\mathrm{MPT}_n$ (multiple plateaus with traps) works with $\lfloor k/5 \rfloor$ traps. The definition of $\mathrm{MPT}_n$ is at first illustrated for five traps.

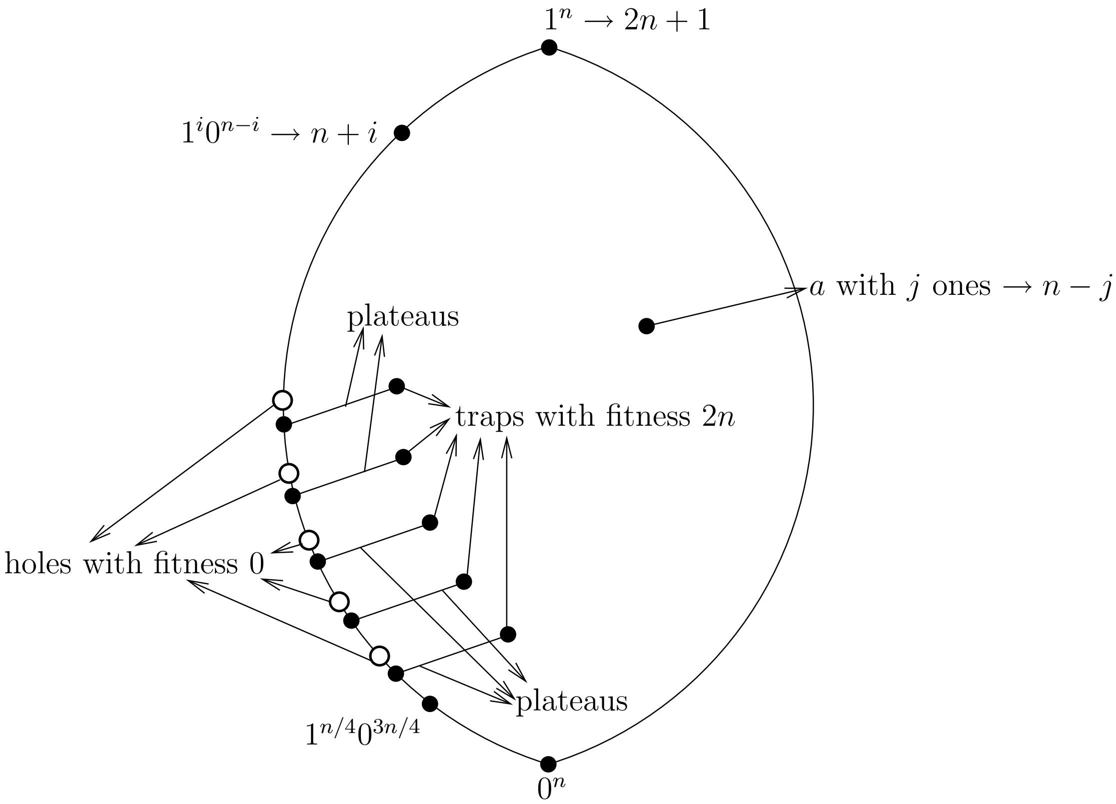



Figure 4: Illustration of the function $\mathrm{MPT}_n$.

**Definition 16.** *The main path $M$ of $MPT_n$ consists of all points $1^i0^{n-i}$ except the $N := \lfloor k/5 \rfloor$ holes. The $j$-th hole $h_j$ is the string $1^{i(j)}0^{n-i(j)}$ where $i(j) = \lfloor n/4 \rfloor + j(k+1) - 1$. The $j$-th plateau $P_j$ consists of the points $1^{i(j)-1}0^{n-i(j)+1-r}1^r$, $1 \le r \le m-1$, and the $j$-th trap $T_j$ consists of the point $t_j^* = 1^{i(j)-1}0^{n-i(j)+1-m}1^m$. The function $MPT_n$ is defined in the following way.*

$$MPT_n(a) = \begin{cases} 2n+1 & \text{if } a = 1^n \\ i & \text{if } a = 1^i0^{n-i} \in M \setminus \{1^n\} \\ 0 & \text{if } a \text{ is a hole} \\ 2n & \text{if } a \text{ is a trap} \\ i(j)-1 & \text{if } a \in P_j \\ n - ONEMAX_n(a) & \text{otherwise.} \end{cases}$$

**Theorem 17.** *The success probability of the $(1+1)^*EA$ on $MPT_n$ within $O(n^3)$ steps is bounded below by $1 - (1/2)^{\Omega\left(n^{1/4}\right)}$.*

*Proof.* We consider a typical run of the $(1+1)^*$EA on $\mathrm{MPT}_n$ and bound the failure probability by $(1/2)^{\Omega(m)}$, $m = \lfloor n^{1/4} \rfloor$. A typical run does not contain a step where at least $k = \lfloor n^{1/2} \rfloor$ bits flip simultaneously (failure probability $O(n^3/k!)$ which is small enough). A typical run starts with an initial string of at most $(3/4)n$ ones. For the fitness function $n - \text{ONEMAX}_n$ the $(1+1)^*$EA reaches at most one string with $j$ ones, $j \leq (3/4)n$. This string is a random one among the strings with $j$ ones. Each level with respect to the number of ones has for $\text{MPT}_n$ at most 2 strings where the fitness differs from $n - \text{ONEMAX}_n$. The probability that such a string is reached on a level $j$, $(1/4)n \leq j \leq (3/4)n$ is bounded by $O\left(n/\binom{n}{n/4}\right)$. Hence, a typical run meets $M \cup P \cup T$ ($P$ is the union of all $P_j$ and $T$ the union of all $T_j$) in some point of $M$ with less than $n/4$ ones. This happens within $O\left(n^2 \log n\right)$ steps (the failure probability is exponentially small).

Since we assume that less than $k$ bits flip simultaneously, the $(1+1)^*$EA considers the traps independently. The probability to get caught within one trap is bounded by Lemma 15 by $(1/n)^{\Omega(m)}$. The probability to get caught in one of the traps is bounded by $O\left(n^{1/2}\right) \cdot (1/n)^{\Omega(m)} = (1/2)^{\Omega(m)}$. The probability that the $(1+1)^*$EA does not pass one of the traps within $O\left(n^{5/2}\right)$ steps (implying a time bound of $O\left(n^3\right)$ to pass all traps) is bounded by Lemma 15 by $O\left(n^{1/2}\right) \cdot (1/2)^{\Omega(m)} = (1/2)^{\Omega(m)}$. Here we assume that we have reached one of the strings on $M$ before the first hole. Besides this, the $(1+1)^*$EA has to pass at most two short paths with increasing fitness values, both of length $O(n)$. The expected time for this is $O\left(n^2\right)$ and the failure probability for $O\left(n^3\right)$ steps is small enough. □

**Lemma 18.** $s(n) = O\left(n^{-1/2}\right)$.

*Proof.* We can use some of the arguments of the proof of Lemma 15, since the $(1+1)$EA and the $(1+1)^*$EA accept the same strings as long as the actual string belongs to $M_1$. Hence, the probability of reaching $M_2$ before $P \cup T \cup P^*$ is bounded by $O\left((1/n)^2\right)$. If $T$ is reached, there is no way back to $M_2$. We can apply Theorem 7 for the analysis of the behavior of the $(1+1)$EA on $P \cup P^* \cup T$. The probability of reaching $T$ within $O\left(n^{3/2}\right)$ steps is bounded below by a positive constant $\varepsilon$ (remember that the length of the plateau is $m+1 = \Theta\left(n^{1/4}\right)$). The probability to reach $M_2$ within $O\left(n^{3/2}\right)$ steps is bounded above by $O\left(n^{-1/2}\right)$. We consider phases of length $O\left(n^{3/2}\right)$ each. With probability $O\left(n^{-1/2}\right)$ we have a success, with probability $\Omega(1)$ we get trapped, and with the remaining probability we start the next phase. The probability that the first success happens in a phase such that we did not get trapped in one of the previous phases is bounded by $O\left(n^{-1/2}\right)$. □

**Theorem 19.** *The success probability of the* $(1+1)$*EA on* $MPT_n$ *within* $n^{n/2}$ *steps is bounded above by* $(1/n)^{\Omega\left(n^{1/2}\right)}$.

*Proof.* We consider a typical run of the $(1+1)$EA. It starts with overwhelming probability with less than $(3/4)n$ ones. On $n - \text{ONEMAX}_n$ it reaches $0^n$ with overwhelming probability within $O\left(n^2 \log n\right)$ steps and, therefore, visits at most $O\left(n^2 \log n\right)$ strings. Using arguments similar to the proof of Theorem 17 it reaches $M$ at some string $1^i 0^{n-i}$ where $i \leq n/4$.

We investigate the next $O\left(n^4\right)$ steps. With overwhelming probability there is no step where at least $k$ bits flip simultaneously. Hence, the traps can be considered independently. The time for the uninterrupted parts of the main path is bounded with overwhelming probability by $O(n^4)$. The time to get trapped or to jump over one hole is with overwhelming probability bounded by $O(n^3)$. Hence, with overwhelming probability the $(1+1)$EA has found the global optimum $1^n$ or has reached a trap within $O(n^4)$ steps. By Lemma 18, the probability of passing all holes without getting trapped is bounded above by $\left(O\left(n^{-1/2}\right)\right)^{n^{1/2}} = (1/n)^{\Omega\left(n^{1/2}\right)}$. For large $n$, each trap has a Hamming distance of at least $n/2 + n^{1/2}$ to $1^n$. Hence, we have to wait for an event whose probability is bounded above by $(1/n)^{n/2+n^{1/2}}$. The probability that such an event happens within $n^{n/2}$ steps is bounded above by $(1/n)^{n^{1/2}}$. $\square$

# 5 Conclusions

We have shown that small details can change the behavior of evolutionary algorithms significantly. The point of time where the success probability becomes significantly larger than 0 may increase from polynomial to exponential and the same can happen for the point of time where the success probability comes very close to 1. A function is presented which is easy to evaluate and to describe and where it is very advantageous to accept only strings with a better fitness. A very simple function, a short path function with constant fitness values on the path has the opposite behavior. This proves that simple evolutionary algorithms can cope with plateaus whose shape is a path of polynomial length. The investigations also provide examples where local performance measures like the quality gain and the progress rate do not contain enough information to predict the global behavior of evolutionary algorithms.

# References


Bäck, T., Fogel, D. B., and Michalewicz, Z. (1997). (Ed.) Handbook of Evolutionary Computation. Oxford University Press, Oxford.

Droste, S., Jansen, T., and Wegener, I. (1998a). A rigorous complexity analysis of the $(1+1)$ evolutionary algorithm for linear functions with Boolean inputs. ICEC 1998, Int. Conf. on Evolutionary Computation, 499–504.

Droste, S., Jansen, T., and Wegener, I. (1998b). On the optimization of unimodal functions with the $(1+1)$ evolutionary algorithm. PPSN V, Parallel Problem Solving from Nature, LNCS 1498, 13–22.

Droste, S., Jansen, T., and Wegener, I. (1999). Perhaps not a free lunch but at least a free appetizer. GECCO 1999, Genetic and Evolutionary Computation Conference, 833–839.

Garnier, J., Kallel, L., and Schoenauer, M. (1999). Rigorous hitting times for binary mutations. Evolutionary Computation 7, 173–203.

Hagerup, T., and Rüb, C. (1989). A guided tour of Chernoff bounds. Information Processing Letters 33, 305–308.

Horn, J., Goldberg, D. E., and Deb, K. (1994). Long path problems. PPSN III, Parallel Problem Solving from Nature, LNCS 866, 149–158.

Menke, R. (1998). Personal communication.

Motwani, R., and Raghavan, P. (1995). Randomized Algorithms. Cambridge University Press, Cambridge.

Mühlenbein, H. (1992). How genetic algorithms really work. I. Mutation and hillclimbing. PPSN II, Parallel Problem Solving from Nature, 15–25.

Rechenberg, I. (1994). Evolutionsstrategie '94. Frommann-Holzboog, Stuttgart.

Rudolph, G. (1997). How mutations and selection solve long path problems in polynomial expected time. Evolutionary Computation 4, 195–205.

Schwefel, H.-P. (1995). Evolution and Optimum Seeking. Wiley, New York.

Wolpert, D. H., and Macready, W. G. (1997). No free lunch theorems for optimization. IEEE Trans. on Evolutionary Computation 1, 67–82.