

On the Choice of the Population Size

Tobias Storch

Dept. Computer Science II, Univ. Dortmund
44221 Dortmund, Germany
tobias.storch@uni-dortmund.de

Abstract. Evolutionary Algorithms (EAs) are population-based randomized optimizers often solving problems quite successfully. Here, the focus is on the possible effects of changing the parent population size. Therefore, new functions are presented where for a simple mutation-based EA even a decrease of the population size by one leads from an efficient optimization to an enormous running time with an overwhelming probability. This is proven rigorously for all feasible population sizes. In order to obtain these results, new methods for the analysis of the EA are developed.

1 Introduction

Evolutionary Algorithms (EAs) are a broad class of general randomized search heuristics. The probably best-known types of EAs are Genetic Algorithms and Evolution Strategies. Their area of application is as huge as their variety and they have been applied successfully in numerous situations. Here, we consider the problem to maximize pseudo-Boolean functions $f_n : \{0, 1\}^n \rightarrow \mathbb{R}_0^+$. We remark that analysis in discrete search spaces differs substantially from that in continuous ones.

With regard to populations, the problems how to choose its size and how to find a method to preserve the diversity are well known. If the size of the population or its diversity are too small, the EA is likely to stagnate in local optima. On the other hand, the EA is likely to waste much time on the evaluation of unnecessary elements, if the population or diversity are too large. Many ideas have been presented to cope with the difficulty of the correct choice of these parameters and they all have shown their usefulness in experiments, e.g., niching methods, multistarts, and many more. In order to understand the success of EAs, theory often investigates the behavior of simple EAs on typical or constructed problems. These artificial problems are often developed to illustrate particular effects of EAs or one of their components at best. Our aim is to illustrate conveniently that the choice of the parent population size may be critical. Therefore, we develop functions where even a decrease of the parent population size by one leads from an efficient optimization with an overwhelming probability to an enormous running time.

We estimate the efficiency of a randomized algorithm. Therefore, let T_{A, f_n} be the random number of function evaluations until algorithm A first evaluates an

optimal search point of f_n . If the expected value of T_{A,f_n} is polynomially bounded in the dimension of the search space n , we call A *efficient* on f_n and *inefficient*, if the expected value of T_{A,f_n} is at least exponentially bounded. Finally, we call A *totally inefficient* on f_n , if after exponential many steps the probability that an optimal search point has been evaluated, remains exponentially small. In this particular situation a polynomially bounded number of (parallel) (independent) multistarts of A is inefficient. Moreover, we are interested in asymptotical results with respect to n .

We investigate one of the best-known EAs. This is the so-called $(\mu+\lambda)$ EA working with a parent population of size $\mu \geq 1$ and an offspring population of size $\lambda \geq 1$. Surprisingly, on many typical functions even the $(1+1)$ EA is quite efficient. Indeed, Jansen and De Jong (2002) considered the role of the offspring population size. They presented functions where a decrease of this parameter leads to enormous differences in the optimization time. Jansen and Wegener (2001b) have shown something less strong for the role of the parent population size. Witt (2003) improved this result.

We develop functions $f_{n,d}$ where the considered mutation-based $(\mu+1)$ EA is totally inefficient, if the parent population has size $\mu \leq d$. However, if the population has size $\mu > d$, the EA is efficient. We introduce such functions for all $d \in \{1, \dots, n^c\}$ and every constant $c > 0$. And we call d the *threshold value of the population size*. In order to prove these results rigorously, we present simple but powerful methods to analyze this EA. They extend the so-called *method of f -based partitions* and help to upper bound the expected optimization time of the $(\mu+1)$ EA on a particular function (see Wegener (2002)).

The paper begins in Section 2 with an introduction of the investigated steady-state $(\mu+1)$ EA. Section 3 presents the desired extensions of the method of f -based partitions and Section 4 exhibit our first results. These results handle only the threshold value of the population size one and do not satisfy all the desired properties but they illustrate some of the main effects which occur. We divide the possible threshold values of the population size d into three domains. For convenience, we consider them in an unnatural order later.

- The first domain encloses $d \in \{1, \dots, \lceil n/(c_1 \log n) \rceil - 1\}$ for some constant $c_1 > 0$. These are investigated in Section 7.
- The second domain encloses $d \in \{\lceil n/(c_1 \log n) \rceil, \dots, \lceil n/c_2 \rceil\}$ for some constant $c_2 > 0$. These are investigated in Section 5.
- And the third domain encloses $d \in \{\lceil n/c_2 \rceil + 1, \dots, n^c\}$ for every constant $c > 0$. These are investigated in Section 6.

We finish with some conclusions.

2 The Steady-State $(\mu+1)$ EA

The considered mutation-based steady-state $(\mu+1)$ EA works with a natural and weak method to preserve diversity. It just avoids duplicates of elements in the population. This technique can be understood as a special niching method. Moreover, in this case the population structure is not only a multiset but a set.

($\mu+1$) EA

1. Choose μ different individuals $x_i \in \{0, 1\}^n$, $1 \leq i \leq \mu$, uniformly at random. These individuals constitute the population \mathcal{P} , i.e., $\mathcal{P} = \{x_1, \dots, x_\mu\}$.
2. Choose an individual x from the population \mathcal{P} uniformly at random. Create y by flipping each bit in x independently with probability $1/n$.
3. If $y \notin \mathcal{P}$, i.e., $y \neq x_i$ for all i , $1 \leq i \leq \mu$, then let $z \in \mathcal{P} \cup \{y\}$ be randomly chosen among those individuals with the worst f -value and let the population be $\mathcal{P} \cup \{y\} - \{z\}$, goto 2., else let the population be \mathcal{P} , goto 2.

Obviously, only populations of size $\mu \leq 2^n$ are possible. In Step 2, the parameter $1/n$ is the standard choice for mutations.

We remark that the theorems of Sections 5, 6 and 7 hold, if fitness-proportional selection (x_i , $1 \leq i \leq \mu$, is chosen with probability $f(x_i)/\sum_{k=1}^\mu f(x_k)$) instead of uniform selection (x_i , $1 \leq i \leq \mu$, is chosen with probability $1/\mu$) is used in Step 2. Furthermore, it is irrelevant which of the elements with smallest f -value is deleted in Step 3.

3 Methods for Upper Bounds on the Expected Optimization Time for the ($\mu+1$) EA

We present two extensions for the ($\mu+1$) EA of the method of f -based partitions (see Wegener (2002)). These extensions can easily be combined. At first, we recall the original method of f -based partitions that is a simple proof technique which helps to upper bound the expected running time of the (1+1) EA to optimize a particular function.

Given $A, B \subseteq \{0, 1\}^n$, $A, B \neq \emptyset$, the relation $A <_f B$ holds, iff $f(a) < f(b)$ for all $a \in A$, $b \in B$ and a pseudo-Boolean function f . Moreover, we call $(A_1, \dots, A_m; f)$ an f -based partition, iff A_1, \dots, A_m is a partition of $\{0, 1\}^n$ and $A_1 <_f \dots <_f A_m$ holds. Furthermore, A_m merely consists of optimal elements a , i.e., $f(a) = \max\{f(b) \mid b \in \{0, 1\}^n\}$. Finally, let $p(a)$, $a \in A_i$, $i < m$, be the probability that a mutation of a creates some $b \in A_{i+1} \cup \dots \cup A_m$ and let $p(A_i) := \min\{p(a) \mid a \in A_i\}$, $i < m$, i.e., $p(A_i)$ constitutes a lower bound on the probability to leave A_i . Given an f -based partition $(A_1, \dots, A_m; f)$, the expected optimization time of the (1+1) EA is bounded above by

$$1 + p(A_1)^{-1} + \dots + p(A_{m-1})^{-1} \quad .$$

Our first extension of this method for the ($\mu+1$) EA allows to disregard areas of the search space at the expense of the population size. It is even not necessary to know the areas, only their sizes. Therefore, let $(A_1, \dots, A_m; f; A)$ be the variant (1) of f -based partitions where A_1, \dots, A_m is only a partition of $A \subseteq \{0, 1\}^n$ but all other constraints still hold.

Theorem 1. *Let $A_0 \subset \{0, 1\}^n$ and $(A_1, \dots, A_m; f; \{0, 1\}^n - A_0)$ be given. The expected optimization time of the ($\mu+1$) EA, where $\mu \geq |A_0| + 1$, is bounded above by*

$$\mu(1 + p(A_1)^{-1} + \dots + p(A_{m-1})^{-1}) \quad .$$

Proof. The initialization evaluates μ different elements. There always exist at least $\mu - |A_0| \geq 1$ individuals in the population which do not belong to A_0 . These are at worst all elements of A_1 after the initialization. Once the $(\mu+1)$ EA has left A_i , $i < m$, this area will never be reached again, i.e., at least one element of the population belongs to $A_{i+1} \cup \dots \cup A_m$. Since the probability to select an individual of the population that belongs to A_i , $i < m$, is lower bounded by $1/\mu$, the expected number of fitness evaluations is bounded by $\mu p(A_i)^{-1}$ until an element of $A_{i+1} \cup \dots \cup A_m$ is created. At worst, this is an element of A_{i+1} . \square

If $A_0 = \emptyset$ and $\mu = 1$, we have the original method for the $(1+1)$ EA.

For the original method it is essential that $A_i <_f A_{i+1}$, $i < m$, holds. We weaken this condition at the expense of the population size. Therefore, let $(A_1, \dots, A_m; f; b_1, \dots, b_{m-1})$, $b_i \geq i + 1$, be the variant (2) of f -based partitions where A_1, \dots, A_m is a partition of $\{0, 1\}^n$ but it just holds $A_i <_f A_{b_i}$, $i < m$. Furthermore, A_m still merely consists of optimal elements.

Theorem 2. $(A_1, \dots, A_m; f; b_1, \dots, b_{m-1})$ is given and let

$$v_i := \sum_{1 \leq j < i, b_j > i} |A_j| \quad \text{for all } i < m \quad .$$

The expected optimization time of the $(\mu+1)$ EA, where $\mu \geq \max\{v_j \mid j < m\} + 1$, is bounded above by

$$\mu(1 + p(A_1)^{-1} + \dots + p(A_{m-1})^{-1}) \quad .$$

Proof. After initialization, the population contains at worst only elements that belong to A_1 . At most the elements of A_j where $j < i$ and $b_j > i$ have an f -value at least as large as the worst element of A_i , $i < m$, but belong to $A_1 \cup \dots \cup A_{i-1}$. By definition, these are at most v_i elements. Therefore and since $\mu \geq \max\{v_j \mid j < m\} + 1 \geq v_i + 1$, once the $(\mu+1)$ EA has reached A_i , $i < m$, this area will never be given up again, i.e., at least one element of the population belongs to $A_i \cup \dots \cup A_m$. The expected number of fitness evaluations is bounded by $\mu p(A_i)^{-1}$ until an element $a \in A_j$, $i + 1 \leq j \leq m$, is created. Since a is either optimal or it holds $\mu \geq v_j + 1$, the element a will be inserted in the successive population. \square

If $b_i = i + 1$, $i < m$, and $\mu = 1$, we have the original method for the $(1+1)$ EA again.

4 First Results

Here, we only consider the threshold value of the population size one. The presented functions in this section do not satisfy all the desired properties. But we make observations that help to identify functions in the following sections where all these properties hold. To obtain these results for all threshold values of the population size, the functions have to be modified in different manners.

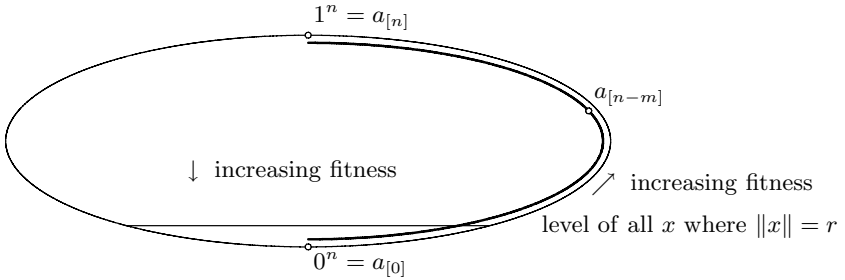


Fig. 1. An illustration of PONEP_n .

Our example functions for the threshold value of the population size one consist of one *global optimum* a_{global} and one *local optimum* a_{local} which is the second-best search point. We call a_{local} a *peak*, too. The *Hamming distance* $\mathbf{H}(x, y)$ of two search points x and y equals the number of indices i where $x_i \neq y_i$. If $\mathbf{H}(x, y) = 1$, we call x and y *Hamming neighbors*. A *path* (of length r) is a sequence of search points $a_{[0]}, \dots, a_{[r-1]}$ where $a_{[i]}$ and $a_{[i+1]}$ are Hamming neighbors and the elements are pairwise distinct. The two search points a_{global} and a_{local} are lying on a path $a_{[0]}, \dots, a_{[n]}$ (of length $n + 1$) that leads from the zero string to the one string. It is $a_{\text{global}} := a_{[n]}$ and $a_{\text{local}} := a_{[n-\lceil n/3 \rceil]}$ and they have (inevitably) Hamming distance $\lceil n/3 \rceil$. The functions have the additional property that with an overwhelming probability the path is first entered in front of a_{local} . Therefore, during a typical run of the investigated EA a_{local} is created before a_{global} is reached. If the population consists only of the second-best search point a_{local} , the probability is extremely small to produce a_{global} , since their Hamming distance is large. But if the population consists of at least one more individual, these elements search forward on the path and find the global optimum efficiently. We remark that the functions are influenced by the short path functions of Jansen and Wegener (2001a).

To define the functions, let 0^k denote the string of k zeros and $|x|$ the number of ones in x . To simplify the notation, let $m := \lceil n/3 \rceil$. Now, we can give a complete definition of PONEP_n (Path with ONE Peak) illustrated in Fig. 1.

$$\text{PONEP}_n(x) := \begin{cases} n + i & \text{if } x = 0^{n-i}1^i =: a_{[i]}, 0 \leq i \leq n \text{ and } i \neq n - m \\ 2n - 1/2 & \text{if } x = 0^m1^{n-m} =: a_{[n-m]} \\ n - |x| & \text{otherwise} \end{cases}$$

We begin our considerations with populations of size $\mu \geq 2$. The result is proven in two different ways to demonstrate different interpretations of the local optimum. The first proof uses Theorem 1 while the second one uses Theorem 2.

Theorem 3. *The expected time until the $(\mu+1)$ EA, where $\mu \geq 2$, has optimized PONEP_n is bounded above by $\mathcal{O}(\mu n^2)$.*

Proof. (using Theorem 1) Let $A_0 := \{a_{[n-m]}\}$ and $(A_1, \dots, A_{2n-1}; \text{PONEP}_n; \{0, 1\}^n - A_0)$ be the variant (1) of f -based partitions where

$$A_i := \begin{cases} \{a \mid \text{PONEP}_n(a) = i\} & \text{if } 1 \leq i < n \\ \{a_{[i-n]}\} & \text{if } n \leq i < 2n - m \\ \{a_{[i-n+1]}\} & \text{if } 2n - m \leq i < 2n \end{cases} .$$

Hence, for the element of A_{2n-m-1} a special 2-bit mutation creates the element of A_{2n-m} and for the elements of A_i , $i < 2n - 1$ and $i \neq 2n - m - 1$, at least one special 1-bit mutation creates an element of A_{i+1} . Therefore, it holds

$$p(A_i) \geq \begin{cases} (1/n^2)(1 - 1/n)^{n-2} \geq 1/(en^2) & \text{if } i = 2n - m - 1 \\ (1/n)(1 - 1/n)^{n-1} \geq 1/(en) & \text{otherwise} \end{cases} .$$

It holds $\mu \geq |A_0| + 1 = 2$. Hence, an application of Theorem 1 leads to an expected optimization time of at most $\mu(1 + en^2 + (2n - 3)en) = \mathcal{O}(\mu n^2)$. \square

Proof. (using Theorem 2) Let $(A_1, \dots, A_{2n}; \text{PONEP}_n; b_1, \dots, b_{2n-1})$ be the variant (2) of f -based partitions where

$$A_i := \begin{cases} \{a \mid \text{PONEP}_n(a) = i\} & \text{if } 1 \leq i < n \\ \{a_{[i-n]}\} & \text{if } n \leq i \leq 2n \end{cases} , \quad b_i := \begin{cases} 2n & \text{if } i = 2n - m \\ i + 1 & \text{otherwise} \end{cases} .$$

Hence, it holds

$$v_i = \begin{cases} |\emptyset| = 0 & \text{if } 1 \leq i \leq 2n - m \\ |A_{2n-m}| = 1 & \text{otherwise} \end{cases}$$

and $p(A_i) \geq 1/(en)$, $1 \leq i < 2n$, since for every element of A_i , $i < 2n$, at least one special 1-bit mutation creates an element of A_{i+1} . It holds $\mu \geq \max\{v_j \mid j < 2n\} + 1 = 2$. Hence, by Theorem 2 the expected optimization time is bounded above by $\mu(1 + (2n - 1)en) = \mathcal{O}(\mu n^2)$. \square

Theorem 4. *With a probability of $1 - \mathcal{O}(1/n)$ the (1+1) EA needs an exponential time $2^{\Omega(n)}$ to optimize PONEP_n . The expected time until the (1+1) EA has optimized PONEP_n is bounded below by $2^{\Omega(n)}$.*

Proof. By Chernoff bounds (see, e.g., Motwani and Raghavan (1995)), the probability is exponentially small that the initial element consists of more than $n - m - \lceil n/7 \rceil$ ones. From then on, at most an element of $\{a \mid |a| \leq n - m - \lceil n/7 \rceil\} \cup \{a_{[0]}, \dots, a_{[n]}\}$ is accepted as population. Each $a_{[i]}$, $i \geq n - m$, consists of at least $n - m$ ones. Hence, a mutation of an element of $\{a \mid |a| \leq n - m - \lceil n/7 \rceil\}$ has to change at least $\lceil n/7 \rceil$ bits to create $a_{[i]}$, $i \geq n - m$. The probability for such a mutation is bounded by $(1/n)^{\lceil n/7 \rceil} = 2^{-\Omega(n)}$. Therefore, when first an element of $\{a_{[n-m]}, \dots, a_{[n]}\}$ is produced this happens with an exponentially small failure probability by a mutation of $a_{[n-m-k]}$, $k \geq 1$. Exactly one k -bit mutation of $a_{[n-m-k]}$ generates $a_{[n-m]}$ and one $(k + l)$ -bit mutation produces $a_{[n-m+l]}$, $1 \leq l \leq m$. The probability of the k -bit mutation equals $(1/n^k)(1 - 1/n)^{n-k} =: q_1$ and of all the $(k + l)$ -bit mutations together $\sum_{l=1}^m (1/n^{k+l})(1 - 1/n)^{n-k-l} =: q_2$. Since $q_2/q_1 = \mathcal{O}(1/n)$, the probability to create $a_{[n-m]}$ before $a_{[i]}$, $i > n - m$, is altogether bounded by $1 - \mathcal{O}(1/n)$. But if $a_{[n-m]}$ is the individual of the population, just a special m -bit mutation generates the global optimum $a_{[n]}$. This is the only element that has an at least as large f -value as the local optimum. The probability for such a mutation is $(1/n)^m(1 - 1/n)^{n-m} = 2^{-\Omega(n)}$. \square

5 Medium Threshold Values of the Population Size

Now, we consider larger threshold values of the population size. Therefore, we enlarge the peak. Of course, as a result, it is not a real peak at all but rather a plateau. We play with the definition of PONEP_n and identify threshold values of the population size that satisfy the desired properties.

Let $\text{PONEP}_{n,d}$, $1 \leq d \leq m := \lceil n/3 \rceil$, be the variant of PONEP_n where the d search points $a_{[n-m]}, \dots, a_{[n-m+d-1]}$ have f -value $2n - 1/2$. Thereby, these elements form the new peak.

Similar to Theorem 3, the expected optimization for the $(\mu+1)$ EA on $\text{PONEP}_{n,d}$, if $\mu \geq d + 1$, is bounded by $\mathcal{O}(\mu n^2)$. In the proof using Theorem 2 it holds $\max\{v_j \mid j < 2n\} + 1 = d + 1$, since an appropriate choice of b_i is now

$$b_i := \begin{cases} 2n & \text{if } 2n - m \leq i \leq 2n - m + d - 1 \\ i + 1 & \text{otherwise} \end{cases} .$$

We investigate the situation $\mu \leq d$ and assume $d \leq \lceil n/c_2 \rceil$ for an appropriate constant $c_2 > 0$. Due to the proof of Theorem 4 the probability is exponentially small that an element of the initial population consists of more than $n - m - \lceil n/7 \rceil$ ones. Furthermore, $q_2/q_1 = \mathcal{O}(1/n^d)$, since we have to compare $(k + l)$ -bit mutations, $0 \leq l < d$, with $(k + d + l)$ -bit mutations, $0 \leq l \leq m - d$. After this, one element of $\{a_{[n-m]}, \dots, a_{[n-m+d-1]}\}$ and some elements of $\{a \mid |a| \leq n - m - \lceil n/7 \rceil\} \cup \{a_{[0]}, \dots, a_{[n-m-1]}\}$ constitute the population. We claim that it typically takes longer to reach the end of the path than to fill up the peak. More precisely, we lower bound the failure probability that within $\lceil 2e\mu n^2/c_2 \rceil$ steps the population consists of some elements of $\{a_{[n-m]}, \dots, a_{[n-m+d-1]}\}$ only. Since we suppress the chance to create $a_{[n]}$ during these steps, we also have to lower bound the probability for this event. Both probabilities are exponentially small. So, with a failure probability of $\max\{\mathcal{O}(1/n^d), 2^{-\Omega(n)}\}$ the $(\mu+1)$ EA, where $\mu \leq d$, needs an exponential time to optimize $\text{PONEP}_{n,d}$.

Since it holds $\mu \leq d$, there exists an element $a_{[n-m+l]} \notin \mathcal{P}$, $0 \leq l < d$, at least until the population consists of elements of $\{a_{[n-m]}, \dots, a_{[n-m+d-1]}\}$ only. A special 1-bit mutation of a correct individual of the population creates an element $a_{[n-m+l]} \notin \mathcal{P}$, $0 \leq l < d$. This element is also inserted into the population. Therefore, we are in a similar situation as in an experiment where the success probability is bounded by $1/(e\mu n)$ in each of $\lceil 2e\mu n^2/c_2 \rceil$ trials. By Chernoff bounds the probability is exponentially small that less than μ , $\mu \leq d \leq \lceil n/c_2 \rceil$, successes occur.

The probability is bounded above by $1/n^j$ to produce $a_{[l+j]}$, $j \geq 0$ and $0 \leq l + j \leq n$, by a mutation of a where $|a| \leq l$. At least j special bits have to change for this event. Hence, let $a_{[n-m+k]} \in \mathcal{P}$, $k \geq 0$, and $a_{[n-m+k+j]} \notin \mathcal{P}$, for all $j \geq 1$. Since the population consists of elements of $\{a \mid |a| \leq n - m - \lceil n/7 \rceil\} \cup \{a_{[0]}, \dots, a_{[n-m+k]}\}$ and $n - m - \lceil n/7 \rceil \leq n - m + k - (\mu - 1)$, there is at most one individual in the population that consists of at most $n - m + k - l$, $0 \leq l \leq \mu - 1$, ones. So, the probability to create $a_{[n-m+k+j]}$ is bounded by $\sum_{l=0}^{\mu-1} 1/(\mu n^{j+l}) \leq 2/(\mu n^j)$. At the beginning $k = d - 1$ holds at best. By Chernoff bounds the probability is exponentially small to create $a_{[n]}$ within $\lceil 2e\mu n^2/c_2 \rceil$ steps, if c_2 is large enough.

Theorem 5. *With a probability of $1 - 2^{-\Omega(n)}$ the $(\mu+1)$ EA, where $\mu \leq d$, needs $2^{\Omega(n)}$ steps to optimize $\text{PONEP}_{n,d}$, $\lceil n/(c_1 \log n) \rceil \leq d \leq \lceil n/c_2 \rceil$ for every constant $c_1 > 0$ and an appropriate constant $c_2 > 0$. The expected optimization time of the $(\mu+1)$ EA, where $\mu \geq d + 1$, is bounded above by $\mathcal{O}(\mu n^2)$.*

6 Large Threshold Values of the Population Size

We consider even larger threshold values of the population size. Therefore, we enlarge the peak more and play more extensively with the definition of PONEP_n . When we investigated the $(\mu+1)$ EA, $\mu \leq d$, on $\text{PONEP}_{n,d}$, we claimed that with an overwhelming probability the peak is filled up before the end of the path is reached. In order to retain this property now, we slow down the arrival at the end of the path, since the peak is larger and thereby, it takes longer to fill it up.

To simplify the notation let $m := \lceil n/3 \rceil$ and $s := \lceil n/c_2 \rceil$ where c_2 is the positive constant of the previous section. Let $\text{PONEP}_{n,d,c}$, $s < d \leq n^c$, for a constant integer $c \geq 1$, be the variant of PONEP_n where $a_{[n-m+s+l]}$, $0 \leq l \leq m - s - 1$ and $l \bmod (c + 1) \neq -1$, have f -value $n - |a_{[n-m+s+l]}|$. Thus, the path behind the peak consists of $\lfloor (m-s)/(c+1) \rfloor$ gaps of size c and possibly one further gap of smaller size. These gaps slow down the arrival at the global optimum. But the path is not a real one at all. Furthermore, beside $a_{[n-m]}, \dots, a_{[n-m+s-1]}$ the elements $a_{(l)}$, $0 < l \leq d - s$, have f -value $2n - 1/2$. Thereby, all these elements form the new peak. But before we describe the appearance of $a_{(l)}$, $1 \leq l \leq d - s$, we remember the Gray Code. The $(\ell$ -digit) *Gray Code* \mathbf{G}_ℓ , $\ell \in \mathbb{R}$, maps the integer x , $0 \leq x \leq 2^\ell - 1$, (bijective) to the binary space $\{0,1\}^\ell$. But in contrast to *Binary Code* the values x and $x + 1$ always have Hamming distance one, $\mathbf{H}(\mathbf{G}_\ell(x), \mathbf{G}_\ell(x + 1)) = 1$ for all $0 \leq x < 2^\ell - 1$. Similar to *Binary Code* $\mathbf{G}_\ell(0) = 0^\ell$ holds. We define $a_{(l)}$. The element $a_{(l)}$, $1 \leq l \leq d - s$, equals $g_{s-1} \dots g_0 0^{m-s} 1^{n-m}$ if $\mathbf{G}_s^{-1}(g_{s-1} \dots g_0) = l$. It holds $l \leq n^c \leq 2^s - 1$, if n is large enough. The mentioned properties of the Gray Code claim that $\mathbf{H}(a_{(l)}, a_{(l+1)}) = 1$, $1 \leq l < d - s$, and $\mathbf{H}(a_{[n-m]}, a_{(1)}) = 1$.

We investigate the situation $\mu \geq d + 1$. An application of Theorem 2 leads to an expected optimization time of $\mathcal{O}(\mu n^{c+2})$. To show this, we define a sequence of the elements of the path and the peak

$$S := (a_{[0]}, \dots, a_{[n-m-1]}, a_{[d-s]}, \dots, a_{(1)}, a_{[n-m]}, a_{[n-m+1]}, \dots, a_{[n-m+s-1]}, a_{[n-m+s-1+(c+1)]}, \dots, a_{[n-m+s-1+\lfloor (m-s)/(c+1) \rfloor (c+1)]}, a_{[n]}) \\ =: (s_0, \dots, s_{n-m+d+\lfloor (m-s)/(c+1) \rfloor})$$

We choose the partition induced by the areas

$$A_i := \begin{cases} \{a \mid \text{PONEP}_{n,d,c}(a) = i\} & \text{if } 1 \leq i < n \\ \{s_{i-n}\} & \text{if } n \leq i \leq 2n - m + d + \lfloor (m - s)/(c + 1) \rfloor \end{cases}$$

and analogously to the previous section for an appropriate choice of the b_i it holds $\max\{v_j \mid j < 2n - m + d + \lfloor (m - s)/(c + 1) \rfloor\} + 1 = d + 1$. Hence, for $i = 2n - m - 1$ we consider the Hamming distance of $s_{i-n} = a_{[n-m-1]}$ and $s_{i+d-s-n} = a_{[n-m]}$ and otherwise of the element of A_i and an appropriate element of A_{i+1} . So,

$$p(A_i) \geq \begin{cases} 1/(en) & \text{if } 1 \leq i < 2n - m + d - 1 \\ 1/(en^{c+1}) & \text{otherwise} \end{cases}$$

Now, we investigate the situation $\mu \leq d$. Similar to the arguments that led to Theorem 5, the probability is exponentially small that an element of $\{a_{[n-m+s-1+k(c+1)]} \mid 1 \leq k \leq \lfloor (m-s)/(c+1) \rfloor\} \cup \{a_{[n]}\}$ is created before an element of the peak $\{a_{[n-m]}, \dots, a_{[n-m+s-1]}, a_{(1)}, \dots, a_{(d-s)}\}$. Furthermore, by Chernoff bounds the failure probability is exponentially small that after $\lceil 2e\mu n^{c+1} \rceil$ steps the population consists of elements with f -value $2n - 1/2$ only. Hence, let $a_{[n-m+s-1+k(c+1)]} \in \mathcal{P}$, $k \geq 1$, and $a_{[n-m+s-1+(k+j)(c+1)]} \notin \mathcal{P}$, for all $j \geq 1$. Since the Hamming distance of $a_{(l)}$, $1 \leq l \leq d-s$, and $a_{[n-m+s-1+(k+j)(c+1)]}$ is at least s , the probability is bounded by $2/(\mu n^{j(c+1)})$ to create $a_{[n-m+s-1+(k+j)(c+1)]}$. By Chernoff bounds the probability is exponentially small that $a_{[n]}$ is created within $\lceil 2e\mu n^{c+1} \rceil$ steps.

Theorem 6. *With a probability of $1 - 2^{-\Omega(n)}$ the $(\mu+1)$ EA, where $\mu \leq d$, needs $2^{\Omega(n)}$ steps to optimize $\text{PONEP}_{n,d,c}$, $\lceil n/c_2 \rceil < d \leq n^c$ for an appropriate constant $c_2 > 0$ and every constant integer $c \geq 1$. The expected optimization time of the $(\mu+1)$ EA, where $\mu \geq d + 1$, is bounded above by $\mathcal{O}(\mu n^{c+2})$.*

7 Small Threshold Values of the Population Size

At first, we consider again the threshold value of the population size one only. But here, the presented functions satisfy the desired properties. After these are proven, we extend our observations up to threshold values of the population size of $\lceil n/(c_1 \log n) \rceil - 1$ for an appropriate constant $c_1 > 0$. Our results for PONEP_n do not satisfy the desired properties, since the probability to jump over the peak is just $\mathcal{O}(1/n)$. We modify PONEP_n that such a situation occurs numerous times. More precisely, our example functions consist of many peaks and paths between them that are also called *bridges*. Together with elements leading to the first peak these form the new path. The global optimum is again located at the end of this path. Typically, the path is first entered in front of the first peak and no shortcuts are taken. This means that never a peak and the bridge located behind it are jumped over. Thus, with an overwhelming probability, at least once a peak is produced before the global optimum is found. Similar to the behavior on PONEP_n , the probability is exponentially small to leave a peak, if the population consists of this peak only. But if there is at least one more individual in the population, these elements search forward on the path and find the next peak efficiently. This goes on until the global optimum is found.

At first, we define the peaks. Therefore, we divide an element x of length n into $\lceil \log n \rceil + 1$ disjoint blocks. Block j , $0 \leq j \leq \lceil \log n \rceil - 1$, encloses the $\lceil n/(4 \log n) \rceil =: s$ bits $x_{js+1}, \dots, x_{(j+1)s}$ and the last block $\lceil \log n \rceil$ the remaining bits $x_{\lceil \log n \rceil s+1}, \dots, x_n$. With block j , $0 \leq j \leq \lceil \log n \rceil - 1$, we associate a bit

$$x_{(j)} := \begin{cases} x_{js+1} & \text{if } x_{js+1} = \dots = x_{(j+1)s} \\ \text{undefined} & \text{otherwise} \end{cases} .$$

Let $a_{[i]}$, $0 \leq i \leq 2^{\lceil \log n \rceil} - 1$, be the element where each $a_{[i](j)}$, $0 \leq j \leq \lceil \log n \rceil - 1$, is not **undefined**, furthermore $\mathbf{G}_{\lceil \log n \rceil}^{-1}(a_{[i](\lceil \log n \rceil - 1)} \cdots a_{[i](0)}) = i$ and block

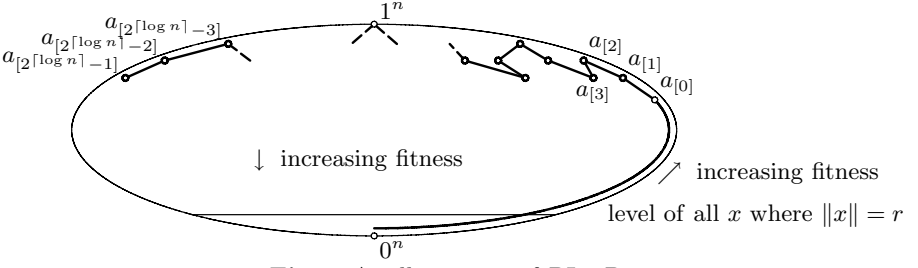


Fig. 2. An illustration of PLINPS_n .

$[\log n]$ consists of ones only. Therefore, in exactly all bits of one block j , $0 \leq j \leq [\log n] - 1$, the elements $a_{[i]}$ and $a_{[i+1]}$, $0 \leq i < 2^{\lceil \log n \rceil} - 1$, differ. The bridge between $a_{[i]}$ and $a_{[i+1]}$ consists of $a_{[i,k]} := a_{[i],1} \cdots a_{[i],js+k} a_{[i+1],js+k+1} \cdots a_{[i+1],n}$, $0 < k < s$. Finally, the elements $a_{[-1,k]} := 0^{n-k} 1^k$, $0 \leq k < n - [\log n]s$, lead from 0^n to the first peak $a_{[0]}$. We remark that the functions are influenced by the long path functions of Rudolph (1997), but ours are short, of course.

$$S := (a_{[-1,0]}, \dots, a_{[-1,n-[\log n]s-1]}, a_{[0]}, a_{[0,1]}, \dots, a_{[0,s-1]}, a_{[1]}, a_{[1,1]}, \dots, a_{[2^{\lceil \log n \rceil - 3}, s-1]}, a_{[2^{\lceil \log n \rceil - 2}]}, a_{[2^{\lceil \log n \rceil - 2}, 1]}, \dots, a_{[2^{\lceil \log n \rceil - 2}, s-1]}, a_{[2^{\lceil \log n \rceil - 1}]}) =: (s_0, \dots, s_{n+s(2^{\lceil \log n \rceil} - [\log n] - 1)})$$

describes the whole path. Now, we can give a complete definition of PLINPS_n (Path with LINEar in n many Peaks) illustrated in Fig. 2.

$$\text{PLINPS}_n(x) := \begin{cases} n + s + i & \text{if } x = s_i \text{ and } x = a_{[j]} \text{ for some } j \\ n + i & \text{if } x = s_i \text{ and } x \neq a_{[j]} \text{ for every } j \\ n - |x| & \text{otherwise} \end{cases}$$

Theorem 7. *The expected time until the $(\mu+1)$ EA, where $\mu \geq 2$, has optimized PLINPS_n is bounded above by $\mathcal{O}(\mu n^3 / \log n)$.*

Proof. The proof is similar to that of Theorem 3 using Theorem 2. We choose the partition induced by the areas

$$A_i := \begin{cases} \{a \mid \text{PLINPS}_n(a) = i\} & \text{if } 1 \leq i < n \\ \{s_{i-n}\} & \text{if } n \leq i \leq 2n + s(2^{\lceil \log n \rceil} - [\log n] - 1) \end{cases} .$$

Hence, it holds $\mu \geq \max\{v_j \mid j < 2n + s(2^{\lceil \log n \rceil} - [\log n] - 1)\} + 1 = 2$ if

$$b_i := \begin{cases} i + 1 & \text{if } i \neq 2n - ([\log n] + j)s \text{ for every } j \\ i + s & \text{if } i = 2n - ([\log n] + j)s \text{ for some } j \end{cases}$$

and $1 \leq i < 2n + s(2^{\lceil \log n \rceil} - [\log n] - 1)$. If we consider the areas A_i and A_{i+1} , it holds $p(A_i) \geq 1/(en)$. So, by Theorem 2 the expected optimization time is bounded above by $\mu(1 + (2n + s(2^{\lceil \log n \rceil} - [\log n] - 1) - 1)en) = \mathcal{O}(\mu n^3 / \log n)$. \square

We consider a technical lemma that summarizes one main property of PLINPS_n .

Lemma 8. For $0 \leq i \leq 2^{\lceil \log n \rceil} - 1$ and all c where $\text{PLINPS}_n(c) > \text{PLINPS}_n(a_{[i]})$ it holds a) $\mathbf{H}(a_{[i]}, c) \geq s$ and b) $\mathbf{H}(a_{[i-1,k]}, c) \geq s$ for arbitrary k .

Proof. The element c can only be $a_{[i+l]}$, $l \geq 1$, or $a_{[i+l,j]}$ for arbitrary j .

- a) The elements $a_{[i]}$ and $a_{[i+l]}$ differ in all bits of at least one block. Therefore, it is $\mathbf{H}(a_{[i]}, a_{[i+l]}) \geq s$ and since by construction of a_i all bits in each block have the same value, it holds $\mathbf{H}(a_{[i]}, a_{[i+l,j]}) \geq \min\{\mathbf{H}(a_{[i]}, a_{[i+l]}), \mathbf{H}(a_{[i]}, a_{[i+l+1]})\} \geq s$.
- b) Due to the situation described in a) it is $\mathbf{H}(a_{[i-1,k]}, a_{[i+l]}) \geq \min\{\mathbf{H}(a_{[i-1]}, a_{[i+l]}), \mathbf{H}(a_{[i]}, a_{[i+l]})\} \geq s$ and it is $\mathbf{H}(a_{[i-1,k]}, a_{[i+l,j]}) \geq \min\{\mathbf{H}(a_{[i-1,k]}, a_{[i+l]}), \mathbf{H}(a_{[i-1,k]}, a_{[i+l+1]})\} \geq \min\{\mathbf{H}(a_{[i-1]}, a_{[i+l]}), \mathbf{H}(a_{[i-1]}, a_{[i+l+1]}), \mathbf{H}(a_{[i]}, a_{[i+l]})\} \geq s$. □

Theorem 9. With a probability of $1 - 2^{-\Omega(n)}$ the (1+1) EA needs $2^{\Omega(n)}$ steps to optimize PLINPS_n .

Proof. When first an element of $\{s_{n-\lceil \log n \rceil s} = a_{[0]}, \dots, s_{n+s(2^{\lceil \log n \rceil} - \lceil \log n \rceil - 1)}\}$ is produced this happens similar to the proof of Theorem 4 with an exponentially small failure probability by a mutation of $a_{[-1, n - \lceil \log n \rceil s - k]}$, $k \geq 1$. We analyze the situation that the population is $a_{[-1, n - \lceil \log n \rceil s - k]}$, $k \geq 1$, or $a_{[i, s - k]}$, $i \geq 0$. By Lemma 8 the probability is bounded by $|S|(1/n)^s = 2^{-\Omega(n)}$ to create an arbitrary element c where $\text{PLINPS}_n(c) > \text{PLINPS}_n(a_{[i+1]})$. Furthermore, again similar to the proof of Theorem 4, the probability to create $a_{[i+1,l]}$ for an arbitrary l before $a_{[i+1]}$ is bounded by $\mathcal{O}(1/n)$. If the population is $a_{[i]}$, $0 \leq i < 2^{\lceil \log n \rceil} - 1$, by Lemma 8 the probability is exponentially small to create an element $c \neq a_{[i]}$ where $\text{PLINPS}_n(c) \geq \text{PLINPS}_n(a_{[i]})$. Hence, the probability to produce the global optimum before an element $a_{[i]}$, $0 \leq i < 2^{\lceil \log n \rceil} - 1$, is bounded by $2^{-\Omega(n)} + \mathcal{O}(1/n)2^{\lceil \log n \rceil - 1} = 2^{-\Omega(n)}$. □

We consider threshold values of the population size of up to $\lceil n/(c_1 \log n) \rceil - 1$ for an appropriate constant $c_1 > 0$. Therefore, we play with the definition of PLINPS_n . This is done similar to the changings of PONEP_n that led to $\text{PONEP}_{n,d}$. We enlarge the peak. Let $\text{PLINPS}_{n,d}$, $1 \leq d < \lceil n/(c_1 \log n) \rceil$, be the variant of PLINPS_n where beside $a_{[i]}$, $0 \leq i < 2^{\lceil \log n \rceil} - 1$, the elements $a_{[i,k]}$, $1 \leq k < d$, have f -value $\text{PLINPS}_n(a_{[i]})$. So, these elements form the new peaks.

The arguments that led to Theorems 5 and 7 bound the expected optimization time for the $(\mu+1)$ EA on $\text{PLINPS}_{n,d}$, if $\mu \geq d + 1$, by $\mathcal{O}(\mu n^3 / \log n)$.

The result of Theorem 9 also holds for the $(\mu+1)$ EA on $\text{PLINPS}_{n,d}$, if $\mu \leq d$ and $d < \lceil n/(c_1 \log n) \rceil$ for an appropriate constant $c_1 > 0$. The path is reached at its beginning. If the population consists only of elements of the peak $\{a_{[i]}, a_{[i,1]}, \dots, a_{[i,d-1]}\}$, $0 \leq i < 2^{\lceil \log n \rceil} - 1$, by Lemma 8 the probability is exponentially small to create an element $c \notin \{a_{[i]}, a_{[i,1]}, \dots, a_{[i,d-1]}\}$ where $\text{PLINPS}_{n,d}(c) \geq \text{PLINPS}_{n,d}(a_{[i]})$, if c_1 is large enough. Otherwise, let $s_k \in \mathcal{P}$ but $s_j \notin \mathcal{P}$ for all $j > k$. If $s_k = a_{[i,l]}$, $l \geq d$, similar to the arguments that led to Theorems 5 and 9, the probability to produce an element of $\{a_{[i+1,d]}, \dots, a_{[i+1,s-1]}\}$ before an element of $\{a_{[i+1]}, a_{[i+1,1]}, \dots, a_{[i+1,d-1]}\}$ is bounded by $\mathcal{O}(1/n^d) =$

$\mathcal{O}(1/n)$. If $s_k = a_{[i]}$ or $s_k = a_{[i,l]}$, $l < d$, the failure probability is bounded by $2^{-\Omega(n/\log n)} = \mathcal{O}(1/n)$ that after $\lceil 2e\mu n^2/(c_1 \log n) \rceil$ steps the population consists of elements of $\{a_{[i]}, a_{[i,1]}, \dots, a_{[i,d-1]}\}$ only. And the probability is also bounded by $\mathcal{O}(1/n)$ that within these steps an element of $\{a_{[i+1]}, a_{[i+1,1]}, \dots, a_{[i+1,s-1]}\}$ is created. Therefore and since for both situations of s_k the probability is exponentially small to produce an arbitrary element c where $\text{PLINPS}_{n,d}(c) > \text{PLINPS}_{n,d}(a_{[i+1]})$, the probability to produce the global optimum before the population consists only of some elements of some peak $\{a_{[i]}, a_{[i,1]}, \dots, a_{[i,d-1]}\}$, $0 \leq i < 2^{\lceil \log n \rceil} - 1$, is again bounded by $2^{-\Omega(n)} + \mathcal{O}(1/n)^{2^{\lceil \log n \rceil} - 1} = 2^{-\Omega(n)}$.

Theorem 10. *With a probability of $1 - 2^{-\Omega(n)}$ the $(\mu+1)$ EA, where $\mu \leq d$, needs $2^{\Omega(n)}$ steps to optimize $\text{PLINPS}_{n,d}$, $1 \leq d < \lceil n/(c_1 \log n) \rceil$ for an appropriate constant $c_1 > 0$. The expected optimization time of the $(\mu+1)$ EA, where $\mu \geq d + 1$, is bounded above by $\mathcal{O}(\mu n^3 / \log n)$.*

Conclusions

We have proved that functions exist where a simple mutation-based EA is efficient if the population size $\mu > d$ and is totally inefficient if $\mu \leq d$. This has been proven rigorously by specifying some functions for all values of d polynomially bounded in the dimension of the search space. These results form a typical so-called hierarchy result. We have developed methods to analyze the investigated EA. These help to upper bound the expected optimization time. The question if the smallest possible increase of the population size may be advantageous has been answered positively. However, in most cases of application such a sensitive decrease of the population size does not have such enormous effects. But these results support the importance of a correct choice of the population size.

Acknowledgements. This research was supported by a Grant from the G.I.F., the German-Israeli Foundation for Scientific Research and Development. The author thanks Ingo Wegener for his help while preparing this paper.

References

- Jansen, T. and De Jong, K. (2002). *An Analysis of the Role of the Offspring Population Size in Evolutionary Algorithms*. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2002, 238–246.
- Jansen, T. and Wegener, I. (2001a). *Evolutionary Algorithms – How to Cope with Plateaus of Constant Fitness and when to Reject Strings with the Same Fitness*. IEEE Transactions on Evolutionary Computation 5, 589–599.
- Jansen, T. and Wegener, I. (2001b). *On the Utility of Populations in Evolutionary Algorithms*. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2001, 1034–1041.
- Motwani, R. and Raghavan, P. (1995). *Randomized Algorithms*. Cambridge University Press, Cambridge.

- Rudolph, G. (1997). *How Mutation and Selection Solve Long-Path Problems in Polynomial Expected Time*. *Evolutionary Computation* 4(2), 195-205.
- Wegener, I. (2002). *Methods for the Analysis of Evolutionary Algorithms on Pseudo-Boolean Functions*. In: Sarker, R., Yao, X. and Mohammadian, M. (Eds.) *Evolutionary Optimization*, 349-369, Kluwer, New York.
- Witt, C. (2003). *Population Size vs. Runtime of a Simple EA*. *Proceedings of the Congress on Evolutionary Computation (CEC) 2003, 1996-2003*.