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

Real Royal Road Functions for Constant Population Size

Tobias Storch and Ingo Wegener

No. CI-167/04

Technical Report ISSN 1433-3325 February 2004

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.

Real Royal Road Functions for Constant Population Size[⋆]

Tobias Storch and Ingo Wegener

Department of Computer Science, University of Dortmund
44221 Dortmund, Germany
{tobias.storch,ingo.wegener}@uni-dortmund.de

Abstract. Evolutionary and genetic algorithms (EAs and GAs) are quite successful randomized function optimizers. This success is mainly based on the interaction of different operators like selection, mutation, and crossover. Since this interaction is still not well understood, one is interested in the analysis of the single operators. Jansen and Wegener (2001a) have described so-called real royal road functions where simple steady-state GAs have a polynomial expected optimization time while the success probability of mutation-based EAs is exponentially small even after an exponential number of steps. This success of the GA is based on the crossover operator *and* a population whose size is moderately increasing with the dimension of the search space. Here new real royal road functions are presented where crossover leads to a small optimization time, although the GA works with the smallest possible population size — namely 2.

1 Introduction and History

Genetic algorithms (GAs) and evolution strategies (ESs) have many areas of application. Here we consider the maximization of pseudo-boolean functions $f_n : \{0,1\}^n \rightarrow \mathbb{R}_0^+$. The success of GAs and ESs depends on the interaction of the different operators, among them the so-called search (or genetic) operators which create new individuals from existing ones. A search operator working on one individual is called mutation and a search operator working on two (or more) individuals is called crossover. We only investigate the best-known crossover operators namely one-point crossover and uniform crossover. There have been long debates whether mutation or crossover is "more important". This paper does not contribute to this debate. Our purpose is to investigate when and why crossover is essential.

The problem of premature convergence and the problem to maintain diversity in the population are well known. There are many ideas how to cope with these problems: multi-starts, fitness sharing, niching, distributed GAs, and many more. They all have shown their usefulness in experiments. It has also been also possible

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

to analyse highly specialized GAs on some functions. However, the rigorous analysis of GAs is still in its infancy. This motivates the investigation of special properties of single operators (like the takeover time).

The success of GAs is based on the use of populations *and* the use of crossover operators. Holland (1975) has discussed why crossover is a good search operator. This has led to the building-block hypothesis and the schema theory (see also Goldberg (1989)). We are interested in a rigorous analysis and we concentrate on the following parameters: $T_{\mathrm{A},f}$ describes the random number of fitness evaluations until the algorithm A evaluates an optimal search point for f. The *expected optimization time* $\mathrm{E}(T_{\mathrm{A},f})$ is the expected value of $T_{\mathrm{A},f}$ and the *success probability function* $t \rightarrow \mathrm{Prob}(T_{\mathrm{A},f} \leq t)$ describes the probability of a successful search within a given number of steps. An algorithm is called *efficient* if the expected optimization time is polynomially bounded (with respect to the problem dimension n) or if at least the success probability within a polynomial number of steps converges to 1 (with respect to n). Crossover is essential for a sequence $f = (f_n)$ of fitness functions if a simple GA (without specialized modules) is efficient while all mutation-based evolutionary algorithms (EAs) are not efficient. Mitchell, Forrest, and Holland (1992) have looked for such functions and presented the so-called royal road functions $\mathrm{RR}_{n,k} : \{0,1\}^n \rightarrow \mathbb{R}_0^+$. The input string a is partitioned to m blocks of length k each where $m = n/k$ is an integer. Then $\mathrm{RR}_{n,k}(a)$ equals the number of blocks of a containing ones only. This is a nice example function since it seems to be a “royal road” for the building-block hypothesis and the application of one-point crossover. However, Mitchell, Holland, and Forrest (1994) (see also Mitchell (1996)) have shown that this intuition is wrong. Crossover is not essential when optimizing the royal road functions. Other “GA-friendly” functions like H-IFF (see, e.g., Watson and Pollack (1999)) have been presented but their analysis is not rigorous. Some rigorous analysis has been performed for highly specialized GAs (see, e.g., Dietzfelbinger, Naudts, van Hoyweghen, and Wegener (2002)). Jansen and Wegener (2001a) have presented so-called real royal road functions for uniform crossover and for one-point crossover and proved rigorously that the expected optimization time of a simple steady-state GA is polynomially bounded while each mutation-based EA needs exponential time until the success probability is not exponentially small. The results hold for populations of size n but not for populations whose size is independent of n.

Hence, the steady-state GA needs a population of moderate size *and* the appropriate crossover operator to be efficient. This raises the question of whether populations without crossover can be essential and the question of whether crossover needs populations whose size grows with n in order to be essential. Jansen and Wegener (2001b) have presented functions where mutation-based EAs working with large populations are efficient while all mutation-based EAs with population size 1 are not efficient. Here we answer the second question by presenting *real royal road functions for populations of size* 2.

More precisely, we describe in Section 2 a steady-state GA (called (2+1)GA) working with the smallest possible population size allowing crossover, namely population size 2. This GA is not specialized. It only guarantees that the population contains two *different* individuals. In Section 3, the real royal road function for uniform crossover and constant population size is presented. It is proven that the (2+1)GA is efficient in the sense, that the success probability after a polynomial number of steps is $1 - o(1)$, i.e., converging to 1 as $n \to \infty$ (Theorem 4). Each mutation-based EA needs exponential time until the success probability is not exponentially small (Proposition 1). In Section 4, the function is changed into a real royal road function for one-point crossover and population size 2. The (2+1)GA needs only a polynomial number of steps to reach a success probability of $1 - o(1)$ (Theorem 6), but the expected optimization time grows exponentially. The reason is that with small probability some bad event happens. This event implies a very large optimization time leading to the large expected optimization time. Therefore, we present in Section 5 a strong real royal road function for one-point crossover and population size 2. For this function the (2+1)GA even has a polynomially bounded expected optimization time (Theorem 11) while mutation-based EAs still need exponential time until the success probability is not exponentially small (Proposition 7). We finish with some conclusions.

2 The Steady-State (2+1)GA

We describe a simple steady-state GA working on the smallest possible population size allowing crossover namely population size 2.

```
(2+1)GA
Initialization: Randomly choose two different individuals $x, y \in \{0,1\}^n$.
Search: Produce an individual $z$, more precisely,
    – with probability $1/3$, $z$ is created by mutate($x$),
    – with probability $1/3$, $z$ is created by mutate($y$),
    – with probability $1/3$, $z$ is created by mutate(crossover($x, y$)).
Selection: Create the new population $\mathcal{P}$.
    – If $z = x$ or $z = y$, then $\mathcal{P} := \{x, y\}$.
    – Otherwise, let $a \in \{x, y, z\}$ be randomly chosen among those individuals
      with the worst $f$-value. Then $\mathcal{P} := \{x, y, z\} - \{a\}$.
```

The reader may wonder why all three possibilities of the search step have probability 1/3. This choice is not essential. Our results hold for all probabilities p_1, p_2, p_3, even if they are based on the fitness of x, y, and z as long as they are bounded below by a positive constant $\epsilon > 0$.

We apply the usual mutation operator flipping each bit independently with probability $1/n$. In Sections 4 and 5, we apply the usual one-point crossover but create only one child, i.e., choose $i \in \{1, \dots, n-1\}$ randomly and set $\text{crossover}(x, y) := (x_1, \dots, x_i, y_{i+1}, \dots, y_n)$. Here the order of x and y is chosen randomly. In Section 3, we apply uniform crossover where each bit of the child is chosen from each of the parents with probability 1/2.

3 Real Royal Roads for the (2+1)GA and Uniform Crossover

Before presenting our new function we emphasize that our purpose is to prove rigorously that the (2+1)GA can outperform all mutation-based EAs. As in many other cases, the first example functions with certain properties are artificial fitness functions designed only to prove the results under considerations.

Our example function has one area of global optima and two local optima of different fitness. It should be difficult to create a globally optimal point by mutation of one of the local optima, but it should be easy to do so by uniform crossover of the two local optima. Hence, the GA has to realize a population consisting of the local optima. The fitness function gives hints to reach the better local optimum first. Since we have always two different individuals, one of them is only close to the local optimum and gets hints to look for the second local optimum. These ideas are now made precise.

To simplify the notation we assume that $m := n/6$ is an even integer. Let $|x|$ be the length of x, $\|x\| := \textsc{OneMax}(x) := x_1 + \cdots + x_n$ denote the number of ones of x, and 0^k a string of k zeros. The Hamming distance $H(x,y)$ equals the number of indices i where $x_i \neq y_i$. A path is a sequence $a_1, \dots, a_p$ such that $H(a_i, a_{i+1}) = 1$ and the points a_i are pairwise distinct. The definition of the new real royal road function $\mathrm{R}_n^{\mathrm{u}}$ (u indicates that we use uniform crossover) is based on a path P and a target region T. The path $P = (a_0, \dots, a_{7m})$ contains $7m + 1$ search points: For $i \leq 6m$ let $a_i := 0^{n-i}1^i$ and for $i = 6m + j$ let $a_i := 1^{n-j}0^j$. Let $\mathrm{R}_n^{\mathrm{u}}(a_i) := n + i$ for all $i \neq 5m$ and $\mathrm{R}_n^{\mathrm{u}}(a_{5m}) := n + 8m$. This implies that we have two local optima on P, namely $a^* := a_{5m}$ and $a^{**} := a_{7m}$. If the (2+1)GA first finds a^*, the second individual can search for a^{**}. Hence, we like to have a good chance of creating an optimal search point by uniform crossover from a^* and a^{**}. Let T contain all points $b1^{4m}c$ where $|b| = |c| = m$ and $\|b\| = \|c\| = m/2$. Uniform crossover between $a^* = 0^m1^{5m}$ and $a^{**} = 1^{5m}0^m$ preserves the $4m$ ones in the middle part. The probability of creating $m/2$ ones in the prefix equals $\binom{m}{m/2}2^{-m} = \Theta(1/m^{1/2})$ (by Stirling's formula). The same holds independently for the suffix. Hence, the probability that uniform crossover on $\{a^*, a^{**}\}$ creates a target point and mutation does not destroy this property equals $\Theta(1/m) = \Theta(1/n)$. Now we give a complete definition of $\mathrm{R}_n^{\mathrm{u}}$ which is illustrated in Figure 1. Let $P_1 := (a_0, \dots, a_{5m-1})$ and $P_2 := (a_{5m+1}, \dots, a_{7m})$.

$$\mathrm{R}_n^{\mathrm{u}}(x) := \begin{cases} 15m & \text{if } x \in T \\ 14m & \text{if } x = a^* \\ 6m + i & \text{if } x = a_i \in P_1 \cup P_2 \\ 6m - \|x\| & \text{if } x \in R := \{0,1\}^n - P - T \end{cases}$$

Proposition 1. *Evolution strategies (without crossover) need with probability $1 - o(1)$ exponentially many steps w.r.t. n to optimize $\mathrm{R}_n^{\mathrm{u}}$.*

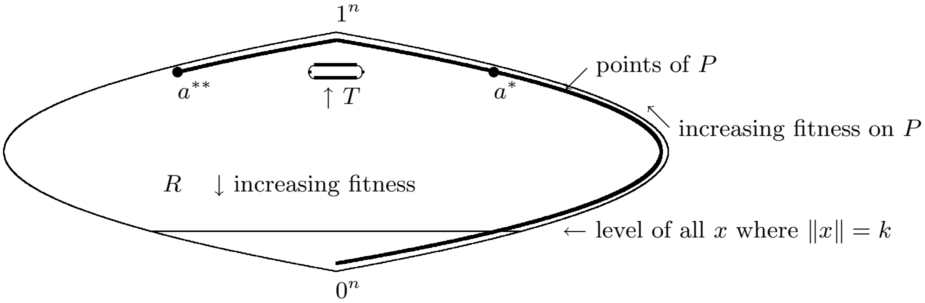
1^n
a^{**}
$\uparrow T$
a^*
points of P
increasing fitness on P
R $\downarrow$ increasing fitness
$\leftarrow$ level of all x where $\|x\| = k$
0^n

Fig. 1. An illustration of $\mathrm{R}_n^{\mathrm{u}}$.

We omit the proof of this proposition. The probability to create a target point by mutation from a path point is exponentially small. Hence, one has to search within R for a small target where the fitness function only gives the advice to decrease the number of zeros. This makes it exponentially unlikely to hit T. A complete proof follows the lines of the proof of Proposition 6 of Jansen and Wegener (2001a).

Lemma 2. *The probability that the* (2+1)GA *does not find a search point from* $P_2 \cup T \cup \{a^*\}$ *within* $c_1 n^2$ *steps is* $2^{-\Omega(n)}$ (c_1 *an appropriate constant*).

Proof. We consider the fitness levels L_i, $1 \leq i < 11m$, containing all search points x where $\mathrm{R}_n^{\mathrm{u}}(x) = i$. These fitness levels contain all search points outside $P_2 \cup T \cup \{a^*\}$. Each of these search points has a better Hamming neighbor. A population $\mathcal{P} = \{x, y\}$ belongs to L_i if $\max\{f(x), f(y)\} = i$. The probability to leave L_i is at least the probability of choosing the right individual for the right 1-bit-mutation and, therefore, at least $p = 1/(3en)$. The steps where crossover is chosen can be ignored. They only may increase the success probability. Hence, we have to wait for at most $11m$ successes in an experiment where each trial has a success probability of at least p. Therefore, the result follows by an application of Chernoff bounds (see, e.g., Motwani and Raghavan (1995)). □

Lemma 3. *If the population contains* a^**, the probability that the* (2+1)GA *does not find an optimal search point, namely a search point from* T*, within* $c_2 n^2$ *steps is* $2^{-\Omega(n)}$ (c_2 *an appropriate constant*).

Proof. By the selection procedure, the population will contain a^* until a search point of T is created. The probability to create some $a \in P_2$ from a^* is bounded below by $p = 1/(3en)$. Afterwards, $\mathcal{P} = \{a^*, a_i\}$, $i > 5m$. We now consider the level defined by the index i until $\mathcal{P} \subseteq \{a^*, a^{**}\} \cup T$. The probability to increase i is bounded below by p in each step. Finally, the probability to create some target point from the population $\{a^*, a^{**}\}$ equals by our arguments above $\Theta(1/n)$. Moreover, only target points are accepted if $\mathcal{P} = \{a^*, a^{**}\}$. Hence, we have to wait for $2m + 1$ successes in experiments where each trial has a success probability of $\Theta(1/n)$. Therefore, Lemma 3 follows by the same arguments as Lemma 2. □

Theorem 4. *The success probability that the* (2+1)GA *with uniform crossover optimizes* $\mathrm{R}_n^{\mathrm{u}}$ *within* cn^2 *steps, c an appropriate constant, is* $1 - O(1/n)$.

Proof. Applying Lemmas 2 and 3 we are left with the problem whether the (2+1)GA creates a search point from P_2 before a search point from $T \cup \{a^*\}$ (called bad event). It is sufficient to bound the probability of the bad event by $O(1/n)$. Here we have to cope with the "undesired" effects of uniform crossover. Remember that all search points of $P_2 \cup T \cup \{a^*\}$ contain $4m$ ones in their "middle part“.

By Chernoff bounds, the probability that the initial population contains a search point with more than $(7/2)m$ ones is exponentially small. As long as no point from $P \cup T$ is created, we only have search points with at most $(7/2)m$ ones if we start with such strings. Hence, each of the search points has at least $m/2$ wrong bits in the middle part. The probability to correct a wrong bit by uniform crossover is at most $1/2$ and the probability to correct it by mutation is $1/n$. Hence, it is very likely to create a search point a_i, $i \leq (15/4)m$, before creating a point from $P_2 \cup T \cup \{a^*\} \cup \{a_i \mid (15/4)m < i < 5m\}$ (the failure probability is exponentially small). We can repeat these arguments to prove that at some point of time the population contains a_i and a_j, $0 \leq i < j < 4m$. Uniform crossover applied to a_i and a_j creates some a_k, $i \leq k \leq j$. Therefore, better points are created by mutation from some a_k, $k < 4m$. The probability of creating some point of $P_2 \cup \{a^*\}$ is exponentially small. Thus, we will obtain in $O(n^2)$ steps a_i and a_j, $4m \leq i < j < 5m$, with overwhelming probability. Then better points are created from some a_k, $4m \leq k < 5m$. In this case, there is exactly one $(5m-k)$-bit mutation to create a^* and exactly one $(5m-k+l)$-bit mutation, $1 \leq l \leq 2m$, to create the point a_{5m+l} from P_2. The probability of the $(5m-k)$-bit mutation equals $q_1 := (1/n)^{5m-k}(1-1/n)^{n-5m+k}$ and the probability of all the $(5m-k+l)$-bit mutations, $1 \leq l \leq 2m$, altogether is the sum q_2 of all $(1/n)^{5m-k+l}(1-1/n)^{n-5m+k-l}$. Since, $q_2/q_1 = O(1/n)$, the probability to create a point from P_2 before creating a point from $T \cup \{a^*\}$ is altogether bounded by $O(1/n)$. □

Having this essential result we can play with the definition of $\mathrm{R}_n^{\mathrm{u}}$. Let $\mathrm{R}_{n,k}^{\mathrm{u}}$ be the variant of $\mathrm{R}_n^{\mathrm{u}}$ where $a_{5m+1}, \ldots, a_{5m+k}$ belong to the region R. Proposition 1 and Lemma 2 hold also for $\mathrm{R}_{n,k}^{\mathrm{u}}$. In Lemma 3, we now have to wait for the event to flip the right $k+1$ bits of a^* to obtain a_{5m+k+1}. The expected time for this is bounded above by en^{k+1}. After $c_2 n^{k+1} \log n$ steps the probability of not creating a point of P_2 is bounded above by

$$\left(1 - \frac{1}{en^{k+1}}\right)^{c_2 n^{k+1} \log n} \leq e^{-(c_2/e)\log n} \quad .$$

This can be made smaller then $1/n^k$ by choosing c_2 appropriately. However, in the proof of Theorem 4 now $q_2/q_1 = O(1/n^k)$, since we need at least $(k+1)$-bit mutations to create points from P_2. This leads to the following result.

Theorem 5. *The success probability that the* (2+1)GA *with uniform crossover optimize* $\mathrm{R}_{n,k}^{\mathrm{u}}$ *within* $cn^{k+1} \log n$ *steps, c an appropriate constant, is* $1 - O(1/n^k)$.

4 A Variant for One-Point Crossover

In order to obtain a real royal road function for the (2+1)GA and one-point crossover we can consider an appropriate variant of $\mathrm{R}_n^{\mathrm{u}}$. The probability of creating 1^n by one-point crossover from $a^{**} = 1^{5m}0^m$ and $a^* = 0^m 1^{5m}$ is bounded below by a positive constant. The idea is to make 1^n the only target point which cannot be reached easily by mutation from a point on P_2. For this reason we replace the path between a^* and a^{**} by a path which is far away from 1^n. The function R_n^1 is defined in the same way as R_n^u with two exceptions. The target set T is replaced by $T := \{1^n\}$. The points $(a_{5m+1}, \ldots, a_{7m})$ are now defined by $a_{5m+2i-1} := 1^i 0^{m-i} 1^{5m-i+1} 0^{i-1}$ and $a_{5m+2i} := 1^i 0^{m-i} 1^{5m-i} 0^i$, $1 \leq i \leq m$. Then the points of P_2 have $5m$ or $5m+1$ ones and are far from 1^n and far from the typical points created in the initialization step. Thus Proposition 1 also holds with the same arguments for R_n^1.

Theorem 6. *The success probability that the* (2+1)GA *with one-point crossover optimizes* R_n^1 *within* cn^2 *steps,* c *an appropriate constant, is* $1 - O(1/n)$.

Proof. It is easy to prove the claims of Lemmas 2 and 3 also for R_n^1. The old proofs can be used without changes. If the population consists of a_i and a_j, $0 \leq i < j < 4m$, we can use the arguments of the proof of Theorem 4 to conclude that the probability of creating a point from P_2 before creating a point from $T \cup \{a^*\}$ is bounded by $O(1/n)$. Let x and y be the search points of the initial population. All search points from P_2 have $4m$ ones in their middle part. By Chernoff bounds, the probability that x and y have not at least $m/2$ common zeros in this middle part is exponentially small. In order to obtain a search point from P_2 it is necessary to flip each of these positions at least once. In a phase of cn steps the probability of not flipping a bit at a special position equals $(1-1/n)^{cn}$ and the probability of flipping it is therefore $1-(1-1/n)^{cn}$ and thus bounded above by a constant $\epsilon < 1$. Since the bit positions are treated independently by the mutation operator, the probability of flipping $m/2$ given positions is exponentially small. (These calculations are related to the coupon collector's theorem (see Motwani and Raghavan (1995)).) If this event does not happen we are in the situation of the fitness function $6m - \|x\| = n - \textsc{OneMax}(x)$. The standard analysis of OneMax leads to the result that we can expect after cn steps a population of two search points of at most m ones. Each step has a constant probability of decreasing the number of ones in the population. By Chernoff bounds, cn steps for an appropriate c are enough to decrease the number of ones from at most $11m$ ones to at most $2m$ ones. The failure probability again is exponentially small. If both search points have at most m ones, crossover can only create a seach point with $2m$ ones and even then the probability of creating $4m$ ones by mutation is exponentially small. Hence, we create some a_i and a_j, $0 \leq i < j < 4m$, before some point from P_2 with a probability exponentially close to 1. □

With the same notations and arguments leading to Theorem 5 we can prove a similar theorem for $\mathrm{R}_{n,k}^1$.

5 Real Royal Roads for the (2+1)GA and One-Point Crossover

The aim is to define a real royal road function R_n^{1*} for the (2+1)GA and one-point crossover which even has the property that the expected optimization time of the (2+1)GA is bounded by some polynomial $p(n)$. This implies by Markoff's inequality a success probability of at least 1/2 within $2p(n)$ steps. Since the bound on the expected optimization time holds for *all* initial populations, the success probability within $O(p(n)\log n)$ steps can be bounded by $1 - O(n^{1/k})$ for each constant k.

The definition of R_n^{1*} is influenced by the function R_n^1. We modify R_n^1 in the following way. All a_i, where $i = 5m + 2j - 1$ and $1 \leq j \leq m$, $i = 5m + 2$ or $i = 5m + 4$, now belong to the bad region R. Finally, all other individuals a_i, $i > 5m$, have the same fitness $13m$ and define a plateau of constant fitness. The full definition of R_n^{1*} is:

$$\mathrm{R}_n^{1*}(x) := \begin{cases} 15m & \text{if } x \in T := \{1^n\} \\ 14m & \text{if } x = 0^m 1^{5m} =: a_0 \\ 13m & \text{if } x \in \{a_i := 1^i 0^{m-i} 1^{4m} 1^{m-i} 0^i \mid 3 \leq i \leq m\} =: P \\ 6m + i & \text{if } x = 0^{n-i} 1^i,\ 0 \leq i < 5m \\ 6m - \|x\| & \text{otherwise} \end{cases}$$

The following result can be proved like Proposition 1.

Proposition 7. *Evolution strategies* (*without crossover*) *need with a probability exponentially close to* 1 *exponentially many steps w.r.t. n to optimize* R_n^{1*}.

Finally, we analyse the (2+1)GA with one-point crossover on R_n^{1*}.

Lemma 8. *The expected time until the population of the* (2+1)GA *contains a search point from $Q := P \cup T \cup \{a_0\}$ is bounded by $O(n^2)$. With a probability of $1 - O(1/n^6)$ this search point is contained in $T \cup \{a_0\}$.*

Proof. As in the proof of Theorem 6 we get an expected time of $O(n^2)$ for creating a search point from Q. Again we have to create with overwhelming probability the search point from Q by mutation of some $0^{n-i}1^i$, $0 \leq i < 5m$. The probability that this happens for some $i \leq 4m$ is exponentially small. Otherwise, $H(0^{n-i}1^i, a_3) = H(0^{n-i}1^i, a_0) + 6$ and we have to compare k-bit mutations with $(k+6)$-bit mutations. Hence, we can apply the same arguments as in the proof of Theorem 5. □

Let us first consider what happens if the population contains only individuals of P. Remember that all these individuals have the same fitness.

Lemma 9. *If the population contains only search points from P, the expected time until the population of the* (2+1)GA *contains an element from $T \cup \{a_0\}$ is bounded by $O(n^9)$.*

	I	II	III	IV	V
a_i	$1\cdots1$	$0\cdots0$	$0\cdots01\cdots1$	$1\cdots1$	$0\cdots0$
a_j	$1\cdots1$	$1\cdots1$	$0\cdots01\cdots1$	$0\cdots0$	$0\cdots0$
a_i	$1\cdots1$	$0\cdots0$	$0\cdots01\cdots1$	$1\cdots1$	$0\cdots0$
length	i	$j-i$	$n-2j$	$j-i$	i

Fig. 2. One-point crossover between a_i and a_j resp. a_j and a_i.

Proof. We only have to consider the situation where the population consists of a_i and a_j, $3 \leq i < j \leq m$. A step is called *essential* if a search point is created, which has a chance of being accepted for the next population, namely a search point from $Q - \{a_i, a_j\}$. The probability of producing $a_{i\pm d}$, $d \geq 1$, $3 \leq i \pm d \leq m$, from a_i by mutation equals $\Theta(1/n^{2d})$, since we consider a special $2d$-bit mutation. In order to investigate mutate(crossover(a_i, a_j)) resp. mutate(crossover(a_j, a_i)) we describe the search points in Figure 2.

If the cut position falls into Region I or Region V (including the borders), we obtain a_i or a_j, altogether each of them with the same probability. If the cut position falls into Region III (including the borders) we obtain $1^i0^{m-i}1^{4m}1^{m-j}0^j$ or $1^j0^{m-j}1^{4m}1^{m-i}0^i$, for Region II we obtain some $1^i0^{s-i}1^{j-s}0^{m-j}1^{4m}1^{m-j}0^j$ or $1^s0^{m-s}1^{4m}1^{m-i}0^i$, $i < s < j$, and for Region IV some $1^i0^{m-i}1^{4m}1^{m-s}0^s$, or $1^j0^{m-j}1^{4m}1^{m-j}0^{j-s}1^{s-i}0^i$, $i < s < j$. The situation is almost symmetric if $3 \leq i < j \leq m$. More precisely, the following individuals have the same chance to be created by crossover and a following mutation from the pair (a_i, a_j):

- a_{i-d} and a_{j+d} (as long as $i - d \geq 3$ and $j + d \leq m$) and
- a_{i+d} and a_{j-d} (as long as $i + d \leq j - d$).

And at least one bit has to flip to obtain an element from $Q - \{a_i, a_j\}$ from crossover(a_i, a_j) resp. crossover(a_j, a_i). Hence, the total probability of an essential step is bounded by $O(1/n)$ and $\Omega(1/n^2)$. In order to prove the lemma it is sufficient to show a bound of $O(n^7)$ on the number of essential steps until a_0 or 1^n is produced.

If $i = 3$, the probability that the next essential step produces a_0 by mutation from a_i is bounded by $\Omega(1/n^5)$. Hence, the expected number of essential steps with a population containing a_3 until a_0 is included in the population is bounded by $O(n^5)$. We are done by proving a bound of $O(n^2)$ on the number of essential steps until the population contains a_3, a_0, or 1^n if we start with a_i and a_j, $3 \leq i < j \leq m$.

For this purpose, it is sufficient to prove that for a phase of cn^2 essential steps, c an appropriate constant, there is a probability of at least a constant $\epsilon > 0$ that we produce a_3, a_0, or 1^n. We ignore the chance of producing a_0 or 1^n. Let $\{a_{i'}, a_{j'}\}$ be the population created from $\{a_i, a_j\}$ in an essential step. The *gain* of this step is defined by $(i' + j') - (i + j)$. To produce a_{i-d} or a_{j+d}, $d \geq 2$, mutation alone has to flip $2d$ bits and the mutation following a crossover at least $2d$ bits. Hence, this happens with a probability of $\Theta(1/n^2)$ in an essential step.

There is a probability of $\left(1-\Theta(1/n^2)\right)^{cn^2} \leq \epsilon'$ where $\epsilon' < 1$ is a constant that the phase does not contain such a step. This implies that $\{a_i, a_j\}$ can create only a_k, $i-1 \leq k \leq j+1$. Further, the probability of good steps of length 1 ($a_i \rightarrow a_{i-1}$, $a_j \rightarrow a_{j-1}$) is the same as the probability of bad steps of length 1 ($a_i \rightarrow a_{i+1}$, $a_j \rightarrow a_{j+1}$). This even holds for steps of length d, $d \geq 2$ ($a_j \rightarrow a_{j-d}, a_i \rightarrow a_{i+d}$). However, if we produce a_j from a_i, we cannot accept the copy of a_j. This does not disturb the symmetry since with the same probability we produce a_i from a_j. If $j = m$, a_{j+1} does not exist. This disturbs the symmetry but supports a positive gain.

The steps of length 1 can be considered as independent coin tosses with success probability 1/2. Applying Chernoff bounds, $\Theta(n^2)$ coin tosses are enough to have a surplus of n wins with a probability of at least a constant $\epsilon'' > 0$. For each fixed point of time the probability that the steps of length d, $d \geq 2$, lead to a non-negative gain is by symmetry at least 1/2 (it is not exactly 1/2, since the total gain of these steps may be 0). This implies that we create a_3 with a probability of at least ϵ in one phase if c is chosen large enough. (Since the search points $a_3, \ldots, a_m$ are a plateau of constant fitness of R_n^{1*}, we have used some ideas contained in the paper of Jansen and Wegener (2001c).) □

Lemma 10. *If the population contains a_0 and some a_j, $3 \leq j \leq m$, the probability for the* (2+1)GA *that the population contains 1^n or a_0 and a_m within the next cn^4 steps, c an appropriate constant, is bounded below by some constant $\epsilon > 0$.*

Proof. We can apply many ideas of the proof of Lemma 9. The probability of creating some a_k, $3 \leq k \leq m$, by mutation of a_0 is bounded by $O(1/n^6)$. Further, the probability of producing $a_{j\pm d}$, $d \geq 2$, by mutation of a_j is bounded by $O(1/n^4)$. Let us investigate the effect of crossover, namely mutate(crossover(a_0, a_j)) and mutate(crossover(a_j, a_0)). We are in the same situation as in Figure 2 but now $i = 0$. Thus, Region I and Region V are empty. If the cut position falls into Region III (including the borders) we obtain $0^m 1^{4m} 1^{m-j} 0^j$ or $1^j 0^{m-j} 1^{4m} 1^m$. To produce $a_{j\pm d}$, $d \geq 1$, the first (or the last) $3 \leq j \pm d \leq m$ bits of the string and d of the last (or the first) m bits must flip by the mutation following crossover. For each of these cut positions, this leads to a probability of $O(1/n^4)$ for a successful crossover. If the cut position falls into Region II, we obtain some $0^s 1^{j-s} 0^{m-j} 1^{4m} 1^{m-j} 0^j$ (II.a) or $1^s 0^{m-s} 1^{4m} 1^m$ (II.b), $0 < s < j$, and for Region IV we obtain some $0^m 1^{4m} 1^{m-s} 0^s$ (IV.a) or $1^j 0^{m-j} 1^{4m} 1^{m-j} 0^{j-s} 1^s$ (IV.b), $0 < s < j$. For (II.a) and (IV.b) we distinguish the cases $s \geq 2$ and $s = 1$.
Case 1. $s \geq 2$. To obtain $a_{j\pm d}$ at least four bits have to flip in the mutation following crossover:

- the first 2 resp. last 2 ones, since $s \geq 2$,
- at least one other *special* bit of the first resp. last m positions, since $d \geq 1$,
- at least $d \geq 1$ *special* bits of the last resp. first m positions.

This again leads for each of these cut positions to a probability of $O(1/n^4)$ for a successful mutate-crossover step.
Case 2. $s = 1$. Here we can guarante only three flipping bits at selected positions. We can use the same arguments but have to take into accout that $s = 1$. However,

the case $s = 1$ refers to only one cut position of crossover. This leads to a probability of $O(1/n^3)$ for a successful mutate-crossover step.
For (II.b) and (IV.a) the mutation following a crossover has to flip $k+|s-k|$ bits to generate a_k. This leads for $s \neq 3$ to a probability of $O(1/n^4)$ and for $s = 3$ to a probability of $O(1/n^3)$ for a successful mutate-crossover step. Therefore, since each cut position has the same probability of $1/(n-1)$, altogether we get a probability of

$$\frac{(n-3) \cdot O(1/n^4) + 2 \cdot O(1/n^3)}{n-1} = O(1/n^4)$$

for creating some a_k. Hence, the probability of a successful crossover or a mutation flipping 4 or even more bits is bounded by $O(1/n^4)$. The probability of including a_m in the population can be analysed in the same way as the probability of including a_3 in the proof of Lemma 9. □

Theorem 11. *The expected time until the* (2+1)GA *has optimized* R_n^{1*} *is bounded above by* $O(n^6)$.

Proof. By Lemmas 8 and 9, an expected number of

$$\bigl(1 - O(1/n^6)\bigr)O(n^2) + O(1/n^6)\max\bigl(O(n^2), O(n^9)\bigr) = O(n^3)$$

steps are enough to obtain a population containing a_0 or 1^n. In the second case we are done. Otherwise, the population will contain a_0 until it has optimized R_n^{1*}. If the second indidividual of the population does not equal some a_j, $3 \leq j \leq m$, a 6-bit mutation will create one from a_0, what needs an expected time of $O(n^6)$. By Lemma 10 the expected time until the population then contains $a_0 = 0^m 1^{5m}$ and $a_m = 1^{5m} 0^m$ is bounded by $O(n^4)$. Afterwards, the probability of creating 1^n is at least $1/(10e)$ (choose crossover, its cut position s in the middle, namely $s \in \{m, \ldots, 5m\}$ and do not mutate any bit). Hence, with a probability of $1-o(1)$ we produce 1^n before a_m is replaced by some a_j, $3 \leq j \leq m-1$. In this case, we can repeat the arguments. The expected number of these phases is $1 + o(1)$. □

We also have defined a real royal road function for uniform crossover where also the expected time of the (2+1)GA is polynomially bounded. The construction of the function and the analysis of the (2+1)GA are more complicated than the results presented here. The reason is the following. If uniform crossover is applied to two search points a and b with a small Hamming distance, the same result can be obtained with a not too small probability by mutating a or b. If the Hamming distance is large, each point which can be created by uniform crossover from a and b has an exponentially small probability of being created. This differs from one-point crossover where each possible search point has a probability of at least $1/(n-1)$ to be created. This implies that uniform crossover can be useful only if there are many good search points "between a and b". This again makes it more difficult to control the "undesired effects" of uniform crossover — in particular, in small populations.

Conclusions

The question whether crossover without special methods to ensure the diversity of the population and without a population whose size grows with the dimension of the search space can improve a mutation-based EA significantly has been solved. Fitness functions have been presented where a simple GA with population size 2 is efficient while mutation-based EAs need with overwhelming probability exponentially many steps. Efficiency is defined as a success probability of $1-o(1)$ within a polynomial number of steps or even as a polynomial expected optimization time. The most important types of crossover, namely uniform crossover and one-point crossover, have been investigated. These are the first results to prove for some examples that crossover can be essential even for populations of size 2. Nevertheless, in most cases of application, it is useful to have larger populations and some method to preserve the diversity in the population.

References

Dietzfelbinger, M., Naudts, B., van Hoyweghen, C., and Wegener, I. (2002). The analysis of a recombinative hill-climber on H-IFF. Submitted for publication in IEEE Trans. on Evolutionary Computation.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison–Wesley, Reading, Mass.

Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems.* The Univ. of Michigan Press, Ann. Arbor, Mich.

Jansen, T. and Wegener, I. (2001a). Real royal road functions — where crossover provably is essential. Proc. of GECCO'2001, 375–382. Morgan Kaufmann, San Francisco, Calif.

Jansen, T. and Wegener, I. (2001b). On the utility of populations in evolutionary algorithms. Proc. of GECCO'2001, 1034–1041. Morgan Kaufmann, San Francisco, Calif.

Jansen, T. and Wegener, I. (2001c). Evolutionary algorithms — how to cope with plateaus of constant fitness and when to reject strings with the same fitness. IEEE Trans. on Evolutionary Computation 5, 589–599.

Mitchell, M. (1996). *An Introduction to Genetic Algorithms.* Chapter 4.2. MIT Press, Cambridge, Mass.

Mitchell, M., Forrest, S., and Holland, J.H. (1992). The royal road for genetic algorithms: Fitness landscapes and GA performance. Proc. of First European Conf. on Artificial Life. MIT Press, Cambridge, Mass.

Mitchell, M., Holland, J.H., and Forrest, S. (1994). When will a genetic algorithm outperform hill-climbing? Advances in NIPS 6, Morgan Kaufmann, San Mateo, Calif.

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

Watson, R.A. and Pollack, J.B. (1999). Hierarchically-consistent test problems for genetic algorithms. Proc. of the Congress on Evolutionary Computation (CEC), 1406–1413. IEEE Press.