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 — Where Crossover Provably is Essential

Ingo Wegener

No. CI-104/01

Technical Report | ISSN 1433-3325 | Junary 2001

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 — Where Crossover Provably is Essential*

Ingo Wegener
FB Informatik, LS 2, Univ. Dortmund, 44221 Dortmund, Germany
wegener@ls2.cs.uni-dortmund.de

Abstract

Mutation and crossover are the main search operators of different variants of evolutionary algorithms. Despite the many discussions on the importance of crossover nobody has proved rigorously for some explicitly defined fitness functions $f_n : \{0,1\}^n \to \mathbb{R}$ that a genetic algorithm with crossover (but without idealization) can optimize f_n in expected polynomial time while all evolution strategies based only on mutation (and selection) need expected exponential time. Here such functions and proofs are presented. For some functions one-point crossover is appropriate while for others uniform crossover is the right choice.

1 Introduction and history

Ideas from biological evolution have influenced the design of systems for various aims, i.e., adaptation, simulation, control, and optimization. Here we consider the optimization, in particular, the maximization of pseudo-boolean functions $f_n : \{0,1\}^n \to \mathbb{R}_0^+$. Evolutionary algorithms use selection operators, search operators, and a stopping criterion. The class of search operators contains mutation where one parent creates one child preferring individuals closer to the parent and crossover where (in most cases) two parents create one or more children which lie in the subcube of $\{0,1\}^n$ spanned by the parents.

There have been long debates which type of search operator is "more important". This paper is not a contribution to this debate. We are interested in specific fitness functions such that crossover is necessary to obtain an evolutionary algorithm (then also called genetic algorithm) where the expected time until an optimal search point is evaluated (called the expected optimization time) is polynomial (instead of exponential).

Holland (1975) has described the possible use of crossover leading to the building-block hypothesis (see also Goldberg (1989)). The well-known

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

schema theorem describes the development of schemata within one step. Hence, in general, it does not lead to results on the expected optimization time. Based on these considerations Mitchell, Forrest, and Holland (1992) have introduced the so-called royal road functions $RR_{n,k} : \{0,1\}^n \rightarrow \mathbb{R}_0^+$ (w.l.o.g. $n = mk$) where the set $\{1, \ldots, n\}$ of indices is partitioned to m consecutive blocks of k elements each. Then $RR_{n,k}(x)$ is defined as the number of blocks containing only x_i-bits equal to 1. Mitchell, Holland, and Forrest (1994) have investigated these functions (for an overview see Mitchell (1996)).

It has turned out that mutation-based evolutionary algorithms are quite successful for the royal road functions. The so-called $(1+1)$EA with population size 1 and mutation probability $1/n$ has an expected optimization time of $O(2^k \cdot \frac{n}{k} \log \frac{n}{k})$. We only mention that we can prove that this bound is asymptotically tight. It has been shown in the above mentioned papers that the expected optimization time of an idealized genetic algorithm (IGA) is of order $2^k \cdot \log \frac{n}{k}$. IGA does not consider the negative implications of the hitchhiking effect. Experiments show that the $(1+1)$EA is faster than genetic algorithms on royal road functions. It is also clear that even the idealized GA saves only a polynomial factor of order n/k. Moreover, the analysis of the royal road functions shows that crossover often has simultaneously positive and negative effects and one has to argue carefully to prove that the positive effects are more important.

Watson, Hornby, and Pollack (1998) and Watson and Pollack (1999) have presented another "GA-friendly" fitness function called H-IFF. For $n = 2^k$ we have 2^m natural blocks of length 2^{k-m} each. The "value" of a block is equal to its length and a block is "activated" if all bits in this block have the same value (0 or 1). Finally, H-IFF(x) is the sum of the values of all activated blocks. The interesting aspect of H-IFF is that the blocks "are strongly and non-linearly dependent on one another" (Watson (2000)). Several aspects of this function have been investigated (Watson, Hornby, and Pollack (1998), Watson and Pollack (1999, 2000a, 2000b), Watson (2000)) where part of the analysis is based on methods due to Wright and Zhao (1999). These papers contain many arguments why mutation-based evolutionary algorithms have exponential expected optimization time while genetic algorithms may have polynomial expected optimization time. However, all analytical results have been obtained under some simplifying assumptions. Nevertheless, the discussion on H-IFF has revealed new aspects of crossover.

The focus of our paper is another one. We are interested in upper and lower bounds on the expected optimization time which are proved without any assumption. The aim is to show that genetic algorithms have on some functions polynomial expected optimization time while mutation-based evolutionary algorithms need exponential expected optimization time. The functions are defined just to have the desired properties (as it was the case with RR and H-IFF). Our functions will not have such a clear "schema structure" as RR and H-IFF. However, our aim is to show that we can control for some "GA-friendly" functions all negative aspects of crossover without using an artificial algorithm. The first paper where the use of uniform crossover has been proved rigorously is by Jansen and Wegener (1998). However, they have used an artificial small crossover probability of $1/(n \log^3 n)$ in order to control the hitchhiking effect and, for their example, the expected optimization time for mutation-based algorithms is only super-polynomial, namely of order $n^{\log n}$, and not exponential.

Since one-point crossover is the historically first crossover operator and since one-point crossover was assumed to be adequate for the royal road functions, we first consider this type of crossover operator. In Section 2, we introduce and analyze the so-called real royal road functions for one-point crossover. In Section 3, we do the same for uniform crossover. We finish with some conclusions.

2 Real royal functions for one-point crossover

Definition 1. *For $x \in \{0,1\}^n$ let $|x|$ be the number of ones in x, i.e., $x_1 + \cdots + x_n$, and let $b(x)$ be the length of the longest block consisting of ones only, i.e., the largest l such that $x_i = x_{i+1} = \cdots = x_{i+l-1} = 1$ for some i. The real royal road functions for one-point crossover are defined by*

$$R_{n,m}(x) = \begin{cases} 2n^2 & \text{if } x = (1,1,\dots,1) \\ n|x| + b(x) & \text{if } |x| \le n - m \\ 0 & \text{otherwise} \end{cases}$$

We use the notation R_n for the special case $m = \lceil n/3 \rceil$.

The function has the property that, as long as $|x| \le n - m$, the fitness depends on the number of ones and ones which build a block are better than ones that are spread over the vector. The all-ones string is optimal and is surrounded by a large valley of bad points. Also H-IFF has the property that the second-best points are far away from the optimal ones. However, for people who like to see "more smooth" functions we can consider such a variant of the real royal road functions R_n where we assume for the ease of description that n is a multiple of 6:

– If $(2/3)n < |x| \leq (5/6)n$, the fitness equals $(10/3)n^2 - 4|x|n + b(x) - |x|$, i.e., the fitness decreases linearly with $|x|$. If $b(x) = |x|$, the fitness decreases from $(2/3)n^2$ to 0.

– If $(5/6)n \leq |x| \leq n$, the fitness equals $12|x|n - 10n^2 + b(x) - |x|$, i.e., the fitness increases linearly with $|x|$. If $b(x) = |x|$, the fitness increases from 0 to $2n^2$.

This variant is for genetic algorithms even easier than the original function for R_n. The lower bounds for evolution strategies without crossover get a bit worse as we show now.

An evolution strategy starts with an initial population of polynomial size. The individuals are chosen randomly and independently. The probability that such an individual has more than $(2/3)n$ ones is bounded above by $e^{-n/18} = e^{-\Omega(n)}$ (application of Chernoff's bound, see Motwani and Raghavan (1995)). Hence, the probability of having an individual with more than $(2/3)n$ ones is exponentially small. If the evolutionary strategy uses a plus-strategy, i.e., accepts only individuals which are not worse than the given ones, the optimal string has to be produced by mutation from an individual with at most $(2/3)n$ ones.

We allow all mutation probabilities where the bits are flipped independently with the same probability $p \leq 1/2$. Then the probability for producing the all-ones string is maximized for strings with $(2/3)n$ ones. The success probability equals $p^{n/3}(1-p)^{2n/3}$ which is maximized for $p = 1/3$ and therefore exponentially small. This implies that the probability of obtaining the optimum in polynomial time or even in time $2^{\varepsilon n}$ for some small $\varepsilon > 0$ is exponentially small. Evolution strategies may allow to accept individuals with more than $(2/3)n$ ones. Then it is the best to have a blind search without any advice, since the optimal string is a single peak like a needle in the haystack. The search region contains exponentially many points and, therefore, the search takes exponential time. For the smooth variant of the real royal road function it may be sufficient to obtain by mutation a point with more than $(5/6)n$ ones from a point with at most $(2/3)n$ ones. The probability for such an event is exponentially small for all mutation probabilities. If we search within the region of more than $(2/3)n$ and less than $(5/6)n$ ones we even get hints to decrease the number of ones. Moreover, the fraction of strings with at least $(5/6)n$ ones among the set of strings with at least $(2/3)n$ ones is exponentially small. This implies the following result.

Proposition 2. *Evolution strategies (without crossover) need with a probability exponentially close to 1 exponentially many steps to optimize the real royal road function R_n (or its smooth variant).*

We now introduce the steady-state GA (genetic algorithm) which we want to analyze. Steady-state GAs are easier to analyze, since we produce only one new individual per step. We use the parameter $s(n)$ for the population size, the parameter $p_c(n)$ for the probability to apply the operator one-point crossover (the two parents are cut after the ith position, $1 \leq i \leq n-1$ is chosen randomly, and the child takes the first i positions from the first parent and the last $n-i$ positions from the second parent), and the standard choice $p_m(n) = 1/n$ for the probability that bits are flipped during mutation.

Algorithm 3. *Steady-state GA*

1.) *Choose independently and randomly the $s(n)$ individuals of the initial population.*

2.) *With probability $p_c(n)$ go to Step $3'$ and with the remaining probability of $1 - p_c(n)$ go to Step $3''$.*

$3'$.) *Choose two parents x and y from the current population. Let z^* be the result of one-point crossover applied to x and y and let z be the result of mutation applied to z^*.*

$3''$.) *Choose one parent x from the current population. Let z be the result of mutation applied to x.*

4.) *If the fitness of z is smaller than the fitness of the worst individual of the current population, go to Step 2. Otherwise, add z to the population. Let W be the multi-set of individuals in the enlarged population which all have the worst fitness and let W' be the set of those individuals in W which have the largest number of copies in W. Eliminate randomly one element in W' from the current population. Go to Step 2.*

Remark. *For the selection procedure in Step $3'$ and $3''$ we only require that $f(x) \geq f(x')$ implies that the probability of choosing x is at least as large as the probability of choosing x' which implies the same selection probabilities for x and x' if $f(x) = f(x')$.*

Algorithm 3 has no stopping criterion, since we want to estimate the expected optimization time. In order to simplify the control of the well-known hitchhiking effect we have introduced a simple and reasonable rule to enlarge the diversity of the population. Among the worst individuals we eliminate one with the largest number of copies.

Theorem 4. *Let $p_c(n) \leq 1 - \varepsilon$ for some $\varepsilon > 0, m \leq \lceil n/3 \rceil$, and $s(n) \geq m + 1$. Then the expected optimization time of the steady-state GA for the* *real royal road functions $R_{n,m}$ is bounded above by $O(n \cdot s(n)^2 \cdot \log s(n) + n^2 \cdot s(n) \cdot m + s(n)^2/p_c(n))$. For the typical case where $p_c(n)$ is a positive constant and $s(n) \leq n$ the bound is $O(n^3(m + \log n))$.*

Proof. We consider several phases of the run of the steady-state GA. Each phase has a goal and we estimate the expected time until the goal is reached.

Phase 1. *The goal is that at least one individual has at most $n-m$ or exactly n ones.*

Claim 1. *The expected time for Phase 1 is bounded by $1 + o(1)$.*

Proof. The probability that the initial population has not the desired property equals by Chernoff's bounds $2^{-\Omega(n^2)}$. The probability that mutation produces an individual with the desired properties is much larger than n^{-n} and the expected waiting time for such an event is at most $n^n = 2^{O(n \log n)}$. Including the initial step we have to wait on average $1 + o(1)$ steps. □

Phase 2. *Phase 1 is finished and the goal is that all individuals have exactly $n - m$ ones or we have found the optimum.*

Claim 2. *The expected time for Phase 2 is bounded by $O(n^2 \cdot s(n)/m)$.*

Proof. We pessimistically assume that we do not find the optimum. Increasing the number of ones is for $R_{n,m}$ more important than to increase the length of the largest 1-block. As long as the individuals do not have all $n - m$ ones, we will show that the probability of increasing the number of ones in the population is at least $\varepsilon \cdot m/(e \cdot n)$ leading to a waiting time of $O(n/m)$. This implies the claim, since it is sufficient to produce $s(n)(n - m)$ ones. We still have to prove the lower bound on the probability of increasing the number of ones in the population. With probability at least ε we only perform mutation. If we choose a parent with less than $n - m$ ones, there are at least m 1-bit mutations increasing the number of ones and each has a probability of $\frac{1}{n}(1 - \frac{1}{n})^{n-1} \geq 1/(e \cdot n)$. If we choose a parent with exactly $n - m$ ones, there is a probability of $(1 - \frac{1}{n})^n \geq 1/e \geq m/(e \cdot n)$ to produce a replica which replaces an individual with less than $n - m$ ones. □

Phase 3. *Phase 2 is finished and the goal is that all individuals x have exactly $n - m$ ones where $b(x) = n - m$ or we have found the optimum.*

Claim 3. *The expected time for Phase 3 is bounded by $O(n \cdot s(n)^2 \cdot \log s(n))$.*

Proof. We pessimistically assume that we do not find the optimum. Then only strings with exactly $n - m$ ones are accepted. Let $b_1 \leq \cdots \leq b_{s(n)}$ be the lengths of the longest 1-blocks of the individuals. Individuals are only replaced with individuals with the same or a larger b-value. Hence, b_1 and $b_1 + \cdots + b_{s(n)}$ are non-decreasing with respect to time. We only consider steps without crossover, since crossover cannot make things worse.

If $b_1 = \cdots = b_{s(n)} = i$, the expected time to obtain an individual with a b-values larger than i is $O(n^2/(n-m-i))$. We may choose any individual. The 1-block with i ones has at least one neighbored 0. There are $n-m-i$ further ones. The 2-bit mutations flipping the neighbored 0 and one of the further $n-m-i$ ones increase the fitness. Each 2-bit mutations has a probability of $(\frac{1}{n})^2(1-\frac{1}{n})^{n-2} \geq 1/(e \cdot n^2)$. The expected waiting time for a good 2-bit mutation is $O(n^2/(n-m-i))$. For each i-value we have to wait once for such an event, $1 \leq i \leq n-m$. Hence, the contribution of such events altogether is $O(n^2 \log n)$.

If $b_1 = i$ and $j > 0$ individuals have a larger b-value, one individual with b-value i is replaced with a better individual if we choose one of the j better individuals and mutation flips no bit. The expected waiting time equals $O(s(n)/j)$. For each of the at most $n-m$ possible i-values we have to consider all values $j \in \{1, \ldots, s(n)-1\}$ leading to the bound $O(n \cdot s(n)^2 \cdot \log s(n))$. Altogether, we have proved Claim 3. □

Phase 4. *Phase 3 is finished and the goal is to obtain a population of individuals containing all possible individuals x with $n-m$ ones and $b(x) = n-m$ at least once or to find the optimum.*

Claim 4. *The expected time for Phase 4 is bounded by $O(n^2 \cdot s(n) \cdot m)$.*

Proof. The number of different second-best optimal strings (i. e., strings x with $|x| = n-m$ and $b(x) = n-m$) equals $m+1$, since the 1-block may start at each of the positions $1, \ldots, m+1$. Here it is essential to have $s(n) \geq m+1$. A second-best individual is of type j if the 1-block starts at position j. If Phase 4 has not been finished, there is some j such that the population contains a type-j individual and no type-$(j-1)$ individual or no type-$(j+1)$ individual. In both cases the probability to choose the type-j individual equals $1/s(n)$, since all individuals have the same fitness. Since crossover cannot have negative effects, we only consider steps without crossover. There is always a 2-bit mutation changing a type-j individual into a type-$(j-1)$ individual (flip the 0 at position $j-1$ and the last one of the block) and also a 2-bit mutation changing a type-j individual into a type-$(j+1)$ individual (flip the first one of the block and the first 0 behind the 1-block). The probability of such a 2-bit mutation is at least $1/(e \cdot n^2)$. In the positive case we obtain a "new" individual with the same fitness as all other individuals. We accept this individual and eliminate one individual which is contained at least twice in the population. Remember that the assumption

$s(n) \geq m+1$ ensures such duplicates. Hence, the expected time to increase the number of different individuals is $O(n^2 \cdot s(n))$. The total expected time of this phase is bounded by $O(n^2 \cdot s(n) \cdot m)$, since the number of different individuals has to be increased at most m times. □

Phase 5. *Phase 4 is finished and the goal is to obtain an optimal individual.*

Claim 5. *The expected time for Phase 5 is bounded by* $O(s(n)^2/p_c(n))$.

Proof. Because of the selection scheme we always have at least one type-1 individual $1^{n-m}0^m$ and one type-$(m+1)$ individual 0^m1^{n-m}. The probability of choosing a crossover step with this pair of individuals is at least $p_c(n)/s(n)^2$. Each crossover position p where $m \leq p \leq n-m$ leads to the child 1^n. The probability of such a position is $\frac{n-2m+1}{n-1} \geq \frac{1}{3}$. Moreover, the probability that mutation does not destroy 1^n is at least $1/e$. Altogether, the waiting time for such a good step is $O(s(n)^2/p_c(n))$. □

The theorem follows by summing up the expected times for all phases. □

We see that 1-blocks are building blocks. However, only 1-blocks at the beginning or at the end of the string are useful to obtain the optimum by 1-point crossover.

We have introduced $R_{n,m}$ as real royal road functions for one-point crossover, since we obtain for $m = \lceil n/3 \rceil$ a trade-off of exponential time for evolution strategies without crossover and polynomial time for our steady-state GA.

We also investigate the steady-state GA where one-point crossover is replaced with uniform crossover (for all positions i independently choose x_i or y_i with probability 1/2). The analysis of the first four phases can be used without changes. The probability that uniform crossover creates 1^n from $1^{n-m}0^m$ and 0^m1^{n-m} equals 2^{-2m}. This is polynomially bounded only if $m = O(\log n)$. If $m = \lfloor \log n \rfloor$, we still get the bound $O(n^4)$ for the expected optimization time (if $p_c(n)$ is a constant and $s(n) \leq n$). However, evolution strategies (with a single individual) need time $\Theta(n^{\lfloor \log n \rfloor})$ in this situation. Hence, we obtain the same trade-off as Jansen and Wegener (1999), but for non-artificial values of $p_c(n)$.

3 Real royal road functions for uniform crossover

Real royal road functions for uniform crossover are harder to design. The reason is that 1-point crossover can only create $n-1$ different children. For uniform crossover of x and y we have two possibilities. If the Hamming distance between x and y is small, also the number of different possible children is small. However, in this situation also mutation can create these children with not too small probability. If the Hamming distance between x and y is large, each possible child has a vanishing probability to be created.

In order to simplify the description we assume that $n = 2m$ and $m = 3k$. The input $x \in \{0,1\}^n$ is described as pair $x = (x', x'')$ where x' and x'' both have length m. Furthermore, $x'' = (x''_1, x''_2, x''_3)$ where x''_1, x''_2, and x''_3 all have length k. We say that $x'' \in C$ (C is a circle) if $x'' \in \{0^i 1^{m-i}, 1^i 0^{m-i} | 0 \le i \le m-1\}$. The circle is a closed path (Hamming distance 1 between neighbored points) of length $2m = n$. We say that $x'' \in T$ (T is the target) if each of the substrings x''_1, x''_2, and x''_3 contains $\lfloor k/2 \rfloor$ ones and $\lceil k/2 \rceil$ zeros. For strings a and b let $H(a,b)$ be the Hamming distance between a and b. For a set of strings B let $H(a,B)$ be the smallest Hamming distance between a and some $b \in B$.

Definition 5. *The real royal road functions for uniform crossover are defined by*

$$R^*_n(x', x'') = \begin{cases} n - H(x'', C) & \text{if } x' \neq 0^m \text{ and } x'' \notin C \\ 2n - H(x', 0^m) & \text{if } x'' \in C \\ 0 & \text{if } x' = 0^m \text{ and } x'' \notin C \cup T \\ 3n & \text{if } x' = 0^m \text{ and } x'' \in T \end{cases}$$

This definition needs some explanation. With overwhelming probability, the initial population contains only individuals where x' is far from 0^m. Then the fitness function gives advice that x'' should be changed into a "circle string". This can be done efficiently with mutations only. It is unlikely to create in this phase a string where $x' = 0^m$. If $x'' \in C$, the fitness increases with decreasing distance of x' to 0^m. Then we will have individuals where $x' = 0^m$ and $x'' \in C$. The steady-state GA will ensure that the population will contain all possible $x'' \in C$ (if the population is large enough). However, we are far from the optimal strings where $x' = 0^m$ and $x'' \in T$. Uniform crossover of $0^m 0^i 1^{m-i}$ and $0^m 1^i 0^{m-i}$ has a good chance to create an optimal string and mutation only cannot do this job efficiently.

We admit that this function is an artificial one, but it is the first one where one can prove that uniform crossover decreases the expected optimization time from exponential to polynomial. As for the real royal road functions for 1-point crossover it is possible to define a "smooth" variant of R^*_n. We omit this technical definition.

The analysis of evolution strategies follows the lines of the corresponding analysis in Section 2. The probability that the initial population of polynomial size contains an individual x where x' has less than $m/3$ ones is exponentially small. As long as $x' \neq 0^m$ and $x'' \notin C$ the search on the first half, namely x', is a search for the needle 0^m in a haystack. Hence, the probability of finding in polynomial time a string where $x' = 0^m$ and $x'' \notin C$ is exponentially small. Afterwards, $x'' \in C$ is better than $x'' \notin C$ with the only exception of $x' = 0^m$ and $x'' \in T$. With small mutation probabilities like $1/n$ it is easy to find strings x where $x' = 0^m$ and $x'' \in C$. With large mutation probabilities we miss the strings where $x' = 0^m$. Hence, we need a mutation from x where $x' = 0^m$ and $x'' \in C$ to some y where $y' = 0^m$ and $y'' \in T$. The minimal Hamming distance between some $x'' \in C$ and some $y'' \in T$ is $\Omega(n)$. This follows easily, since two of the three strings x_1'', x_2'', and x_3'' are of type 0^k or 1^k. Hence, we need a mutation step where none of the first m bits flips and a constant fraction of the last m bits flips. The last event has an exponentially small probability if the mutation probability decreases with n. For larger mutation probabilities the first event has an exponentially small probability. This implies the following result.

Proposition 6. *Evolution strategies (without crossover) need with a probability exponentially close to 1 exponentially many steps to optimize the real royal road function R_n^* (or its smooth variant).*

Theorem 7. *Let $p_c(n)$ be some positive constant smaller than 1 and $s(n) \geq n$. Then the expected optimization time of the steady-state GA for the real royal road function R_n^* for uniform crossover is bounded above by $O(n^2 \cdot s(n))$ which is $O(n^3)$ if $s(n) = O(n)$.*

Proof. We follow the same proof strategy as in the proof of Theorem 4.

Phase 1. *The goal is that $x'' \in C$ for all individuals $x = (x', x'')$ of the population or we have found the optimum.*

Claim 1. *The expected time for Phase 1 is bounded by $O(n^2 \cdot s(n))$.*

Proof. We pessimistically assume that we do not find the optimum. As long as there is an individual $x = (x', x'')$ where $x' = 0^m$ and $x'' \notin C \cup T$ (these are the only strings with fitness 0), one of them is eliminated with a positive constant probability. We only consider steps without crossover. Either we choose one of the described individuals. Then it is sufficient that at least one of the first $m = n/2$ bits flips. Otherwise, it is sufficient to construct a replica. Hence, on the average, after $O(s(n))$ steps we have no individual with fitness 0.

Afterwards, we like to eliminate the individuals where $x' \neq 0^m$ and $x'' \notin C$. The "distance" of the population to our goal is measured as the sum of all $H(x'', C)$ where $x = (x', x'')$ belongs to the population, $x' \neq 0^m$, and $x'' \notin C$.

This distance is smaller than $s(n) \cdot m$ and the goal is to decrease it to 0. Because of our selection procedure the distance cannot increase. Therefore, it is sufficient to consider steps without crossover. If we select an individual x, where $x' \neq 0^m$ and $x'' \notin C$, for mutation, there is at least one 1-bit mutation which creates an individual z where $H(z'', C) < H(x'', C)$. If we select an individual where $x'' \in C$, the distance of the population decreases if mutation creates a replica. Hence, the expected waiting time to decrease the distance is $O(n)$ (this is the waiting time for a special 1-bit mutation). We have to wait for such an event for at most $s(n) \cdot m$ times which proves Claim 1. (The bound of Claim 1 can be improved, since often there are many good 1-bit mutations. However, this will not improve the bound of the theorem.) □

Phase 2. *Phase 1 is finished and the goal is that $x' = 0^m$ and $x'' \in C$ for all individuals $x = (x', x'')$ of the population or we have found the optimum.*

Claim 2. *The expected time for Phase 2 is bounded by $O(n^2 \cdot s(n))$.*

Proof. We pessimistically assume that we do not find the optimum. Hence, we only have to consider individuals $x = (x', x'')$ where $x'' \in C$. Now the "distance" of the population to the goal is measured as the sum of all $H(x', 0^m)$ where $x = (x', x'')$ belongs to the population. The distance is at most $s(n) \cdot m$ and the goal is to decrease it to 0. The situation is similar to the proof of Claim 1. The distance does not increase and we consider only steps without crossover. If we select an individual x where $x' \neq 0^m$ for mutation, there is at least one 1-bit mutation which creates z where $H(z', 0^m) < H(x', 0^m)$. Otherwise, $x' = 0^m$. If mutation creates a replica and the distance of the population is positive, we decrease the distance. Altogether, we have proved Claim 2 (and also the bound in Claim 2 can be improved). □

Phase 3. *Phase 2 is finished and the goal is to obtain a population containing all possible individuals x where $x' = 0^m$ and $x'' \in C$ at least once or to find the optimum.*

Claim 3. *The expected time for Phase 3 is bounded by $O(n^2 \cdot s(n))$.*

Proof. We pessimistically assume that we do not find the optimum. Then the population only contains individuals x where $x' = 0^m$ and $x'' \in C$. The circle C is a closed path where each point has two neighbors with Hamming distance 1. As long as the goal is not reached, the population contains at least two individuals, say $x = (x', x'')$ and $y = (y', y'')$, such that $x' = y' = 0^m, x'', y'' \in C$, and both individuals have a Hamming neighbor, say

$\tilde{x} = (\tilde{x}', \tilde{x}'')$ and $\tilde{y} = (\tilde{y}', \tilde{y}'')$ resp., such that $\tilde{x}' = \tilde{y}' = 0^m, \tilde{x}'', \tilde{y}'' \in C$, and $\tilde{x}$ and $\tilde{y}$ do not belong to the current population.

We define the "distance" of the current population to the goal as the number of individuals $z = (z', z''), z' = 0^m$, and $z'' \in C$, which are not contained in the population. The distance is at most $n - 1$ and the goal is to decrease it to 0. Our selection procedure implies that the distance cannot increase. Hence, we look for the expected time to decrease the distance. This happens if we choose a step without crossover, select one of the individuals described above, and perform the "good" 1-bit mutation. Here we need the assumption that $s(n) \geq n = |C|$. Hence, the expected waiting time to decrease the distance is bounded by $O(n \cdot s(n))$ which proves Claim 3. $\square$

Phase 4. *Phase 3 is finished and the goal is to obtain an optimal individual.*

Claim 4. *The expected time for Phase 4 is bounded by $O(n^{3/2} \cdot s(n))$.*

Proof. Because of our selection procedure we always have all individuals $x = (x', x''), x' = 0^m, x'' \in C$, in our population if we have not found the optimum. Here we only consider steps with crossover. Let $x = (x', x'')$ be the first chosen parent. Then the probability of choosing $y = (y', y'')$ where $y' = 0^m$ and $y''_i = 1 - x''_i$ for all i is at least $1/s(n)$. The reason is that y is contained in the population and that all individuals of the population have the same fitness and, therefore, the same chance of being chosen. Let $\tilde{z}$ be the result of uniform crossover applied to x and y. Then $\tilde{z}' = 0^m$ and $\tilde{z}''$ is a random string, since x'' and y'' have different bits at all positions. We have $\tilde{z}'' = (\tilde{z}''_1, \tilde{z}''_2, \tilde{z}''_3)$. The probability that $\tilde{z}''_j, 1 \leq j \leq 3$, contains exactly $\lfloor k/2 \rfloor$ ones and $\lceil k/2 \rceil$ zeros is $\Theta(k^{-1/2})$ (the usual estimate of $\binom{k}{\lceil k/2 \rceil} 2^{-k}$ by Stirling's formula). Hence, the probability that $\tilde{z}'' \in T$ is $\Theta(k^{-3/2})$. Finally, there is a probability of at least $1/e$ that mutation does not destroy $\tilde{z}$. Hence, the success probability is at least $\Omega(s(n)^{-1} \cdot k^{-3/2}) = \Omega(s(n)^{-1} \cdot n^{-3/2})$ and the expected waiting time for a success is bounded by $O(n^{3/2} \cdot s(n))$. $\square$

The theorem follows by summing up the expected times for all phases. $\square$

4 Conclusions

We have presented for the first time functions where it can be proved without any assumption that evolution strategies without crossover need with overwhelming probability exponential time to find the optimum while a realistic steady-state GA has a polynomial expected optimization time. One-point crossover is successful for a function with building blocks. However, the example function has the property that only two of the building blocks are useful to create the optimum by crossover. The real royal road function where uniform crossover works has no building blocks. Here it is essential that the population contains quite different individuals and that it is possible to create individuals "in the middle of the population".

References

[1] Forrest, S., and Mitchell, M. (1993). Relative building-block fitness and the building-block hypothesis. 2. Workshop Foundations of Genetic Algorithms (FOGA), Morgan Kaufmann, San Mateo, Calif.

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

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

[4] Jansen, T., and Wegener, I. (1999). On the analysis of evolutionary algorithms – a proof that crossover really can help. 7. Europ. Symp. on Algorithms (ESA), LNCS 1642, 184-193, Springer, Berlin.

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

[6] Mitchell, M., Forrest, S., and Holland, J. H. (1992). The royal road for genetic algorithms: Fitness landscapes and GA performance. In F. J. Varela and P. Bourgine (Eds.): *Toward a Practice of Autonomous Systems.* Proc. of the First European Conf. on Artificial Life, MIT Press, Cambridge, Mass.

[7] 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.

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

[9] Watson, R. A. (2000). Analysis of recombinative algorithms on a non-separable building-block problem. 5. Workshop Foundations of Genetic Algorithms (FOGA), to appear.

[10] Watson, R. A., Hornby, G. S., and Pollack, J. B. (1998). Modeling building-block interdependency. 5. Symp. Parallel Problem Solving from Nature (PPSN), LNCS 1998, 97-106, Springer, Berlin.

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

[12] Watson, R. A., and Pollack, J. B. (2000a). Symbiotic combination as an alternative to sexual recombination in genetic algorithms. 6. Symp. Parallel Problem Solving from Nature (PPSN). LNCS 1917, 425-434, Springer, Berlin.

[13] Watson, R. A., and Pollack, J. B. (2000b). Recombination without respect: schema disruption in genetic algroithm crossover. Proc. of the Genetic and Evolutionary Computation Conference (GECCO), 112-119, Morgan Kaufmann, San Mateo, Calif.

[14] Wright, A. H., and Zhao, Y. (1999). Markov chain models of genetic algorithms. Proc. of the Genetic and Evolutionary Computation Conference (GECCO), 734-741, Morgan Kaufmann, SanMateo, Calif.