On the Robustness of Evolving Populations

Tobias Friedrich, Timo Kötzing, and Andrew M. Sutton(✉)

Hasso Plattner Institute, Potsdam, Germany
andrew.sutton@hpi.de

Abstract. Most theoretical work that studies the benefit of recombination focuses on the ability of crossover to speed up optimization time on specific search problems. In this paper, we take a slightly different perspective and investigate recombination in the context of evolving solutions that exhibit *mutational robustness*, i.e., they display insensitivity to small perturbations. Various models in population genetics have demonstrated that increasing the effective recombination rate promotes the evolution of robustness. We show this result also holds in the context of evolutionary computation by rigorously proving crossover promotes the evolution of robust solutions in the standard $(\mu + 1)$ GA. Surprisingly, our results show that this effect is still present even when robust solutions are at a selective disadvantage due to lower fitness values.

1 Introduction

The role of crossover in evolutionary computation is still a major open problem in the theory of evolutionary algorithms. In some cases, it can be provably helpful for optimization obtaining quantifiable speed-ups on functions like Jump and OneMax, and particular combinatorial optimization problems on graphs [2,5,7,10,11]. In other cases, recombination can actually be seen as a destructive operator that is detrimental to optimization [13]. The goal of this work is to contribute to our understanding of environments in which crossover can be helpful.

In population genetics, an increased recombination rate has been shown to increase *mutational robustness*: the resistance of fitness to mutational perturbations [3]. In this paper, we want to examine this effect in the context of runtime analysis for evolutionary algorithms. In particular, we introduce a model landscape for which we prove crossover favors regions of higher neutrality. This effect can be seen even when robust solutions have a much weaker fitness gradient than non-robust solutions. On the other hand, as the recombination rate is tuned to zero, greedy hill-climbing behavior takes over and favors regions with sharper fitness gradients: even when these regions contain solutions that are not as robust to perturbations.

Our model landscape is motivated by the fact that in some optimization problems, there could be *sensitive* decision variables that correspond to non-robust solutions and *non-sensitive* decision variables that induce more robust solutions because they correspond to a large plateau. In practice, we will not

J. Handl et al. (Eds.): PPSN XIV 2016, LNCS 9921, pp. 771–781, 2016.
DOI: 10.1007/978-3-319-45823-6_72

know where these sensitive and non-sensitive variables are. However, our results suggest that if they exist, then using crossover tends to favor these more robust solutions even when they are at a fitness disadvantage.

1.1 Visualizing the Evolution of Robustness

We begin with a simple visualization that gives an intuition for why high crossover rates favor robust solutions. The purpose of this short section is to gain some geometric insight into the proofs contained in Sects. 2 and 3. We consider an extension of the cycle $\mathbb{Z}_n$ to a cylinder (formally $\mathbb{Z}_n^2$, but without the added wrap-around in the second dimension). We consider all individuals at the "bottom" of the cylinder to have a small fitness of, say, 5. Fitness grows "upward", but only on certain paths (i.e., for certain x-values). We consider the case of one wide path and many narrow paths. Figure 1 depicts this fitness landscape (darker colors indicate higher fitness). Maximal fitness can be achieved for all maximal y values. We assume that fitness grows more quickly along narrow paths, by a factor of 1.1. Individuals not on any paths with x-value larger than 0 are dead.

To formulate a simple illustrative example of how recombination can favor robust optimization we present the following short experiment (details omitted due to space constraints) to motivate the rest of the paper. We construct a cylinder of size $10^2 \times 10^4$; the wide path has a width of 10 and there are 14 evenly-spaced narrow paths. We consider a $(\mu + \lambda)$ GA with $\mu = 50$ and $\lambda = 250$ evolving on this fitness landscape. During a run, populations with high crossover rates favor the wide path, focusing on the middle of the wide path. On the other hand, mainly asexually reproducing populations favor narrow paths.

Now consider the case where the fitness landscape changes dynamically, but rarely. More precisely, we consider a random shift of the complete fitness landscape in x-direction by ±2 after 6000 iterations, simulating a small but

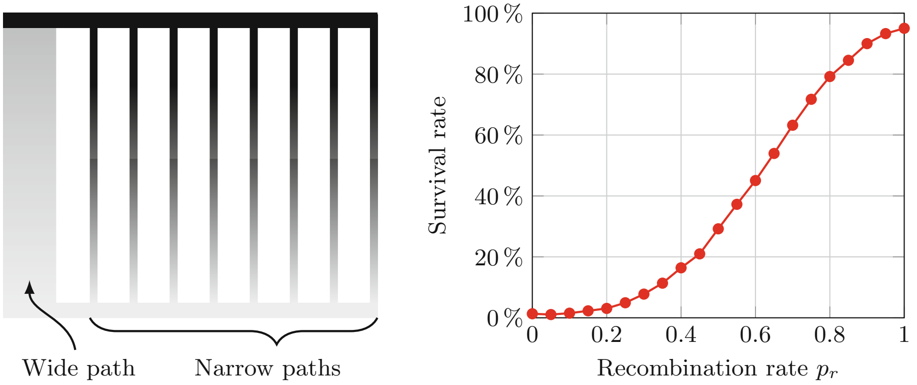

Fig. 1. Left: landscape with many paths leading to the global optimum on the cylinder. Right: prob. GA w/recombination rate p_r does not get extinct during optimization.

significant change in the environment. Populations that exclusively focus on climbing narrow paths will go extinct when the dynamic change of the fitness landscape occurs; populations not too close to the edge of the wide path will survive.

We plot the ratio of populations going extinct before reaching the top y-value, in dependence on the crossover rate p_r (Fig. 1, right). As we can see, with growing crossover rate, it is more and more likely that a population survives and reaches the optimum, due to choosing the wide path to walk up instead of the narrow paths. In particular, an asexually reproducing population will go extinct with high probability, while a population employing crossover in every iteration will go extinct with only very low probability.

2 Preliminaries

We now turn to a more formal analysis in order to prove rigorous statements about how robustness can evolve using recombination. Our aim is to construct a pseudo-Boolean function $f\colon \{0,1\}^n \to \mathbb{R}$ that is structurally similar to the landscape in Fig. 1, but is somewhat easier to work with mathematically. In particular, we want to define a function that has a set of solutions corresponding to a “wide path”, on which each level has an exponential number of solutions, and a collection of sets corresponding to “narrow paths” where there is only a small plateau (each fitness level has only a polynomial number of solutions). We also require the fitness values of the wide path to have a gradual slope, whereas the fitness values of narrow paths have a sharper slope.

Let $n = 2^k$ for some $k \in \mathbb{N}$. We partition a bitstring $x \in \{0,1\}^n$ into three consecutive segments of length k, length $n/2$, and length $(n/2 - k)$.

$$\underbrace{x_1 \dots x_k}_{\text{first segment}}\ \underbrace{x_{k+1} \dots x_{k+n/2}}_{\text{second segment}}\underbrace{x_{k+n/2+1} \dots x_n}_{\text{third segment}}.$$

Denote $[x]_1 = (x_1 x_2 \cdots x_k)$ as the length-k string corresponding to the first segment substring of x. Similarly, $[x]_2$ and $[x]_3$ are the length-$n/2$ and length-$(n/2-k)$ strings formed from the second and third segments of x.

We say that a bit string is *on the wide path* iff the first segment is all 0s. We fix a set $H = \{h_1, h_2, \dots, h_{n/2}\}$ of $n/2$ unique bit strings of length $n/2$. We say a bitstring is *on a narrow path* if (1) it contains at least $\log k$ 1s in the first segment, and (2) the substring in its second segment belongs to H. The first condition ensures adequate separation from the wide path; the second condition defines a collection of subspaces of $\{0,1\}^n$ that contain each narrow path. Formally, we define the set $\mathcal{W}$ of wide-path solutions as $\mathcal{W} = \{x \in \{0,1\}^n \colon x_1 = \cdots = x_k = 0\}$ and the set $\mathcal{N}$ of narrow-path solutions as $\mathcal{N} = \{x \in \{0,1\}^n \colon x_1 + \cdots + x_k \geq \log k \wedge [x]_2 \in H\}$. Each narrow path is associated with a unique $h_i \in H$.

To provide a concrete definition of the set H of narrow path keys, we employ the concept of Hadamard codes [1]. Our motivation is that Hadamard codes provide a clean way of ensuring the narrow paths are sufficiently distant from one another, while simplifying many of the proofs.

A Hadamard code is an error-correcting linear code over binary strings. For our setting, we consider the set of dimension $k-1$ Hadamard codes of length $2^{k-1} = n/2$. To construct this code, we define the inner product between two length $k-1$ bitstrings x and y as $\langle x, y\rangle = \sum_{i=1}^{k-1} x_i y_i \pmod 2$. Let $\sigma\colon \{1,\dots,2^{k-1}\} \to \{0,1\}^{k-1}$ be a bijection. Then, for each $i \in \{1,\dots,2^{k-1}\}$, the i-th codeword is the string $h_i \in \{0,1\}^{2^{k-1}}$ such that $h_{ij} = \langle \sigma(i), \sigma(j)\rangle$ for all $j \in \{1,\dots,2^{k-1}\}$. There are hence $2^{k-1} = n/2$ unique codewords of length $n/2$ and we set $H = \{h_i \mid i \in \{1,\dots,2^{k-1}\}\}$. We explicitly rely on the *minimum distance* property of a dimension $k-1$ Hadamard code: each pair of distinct codewords is separated by Hamming distance at least $2^{k-1}/2$.

We define a pseudo-Boolean fitness function f in such a way so that there is a steeper fitness gradient on the narrow paths. Let $c > 1$ be a constant.

$$f(x) = \begin{cases} \textsc{LeadingOnes}([x]_3), & x \in \mathcal{W}; \\ c\textsc{LeadingOnes}([x]_3), & x \in \mathcal{N}; \\ -\infty, & \text{otherwise.} \end{cases} \tag{1}$$

Here $\textsc{LeadingOnes}(x) := \sum_{i=1}^{n} \prod_{j=1}^{i} x_j$ counts the number of leading ones of its argument. We say an individual is *non-viable* if its fitness is negative infinity. Such an individual corresponds to an *infeasible* solution.

The uniform crossover of two individuals on the wide path always results in an individual on the wide path (with fitness at least the minimal of the two parent fitnesses). On the other hand, uniform crossover of two individuals on separate narrow paths will very likely be non-viable, as we now see.

Lemma 1. *Let $x, y \in H \subseteq \{0,1\}^{n/2}$ such that $x \neq y$. Define $0 < \epsilon < 1$ to be an arbitrary constant. Then with probability $1 - 2^{-\Omega(n)}$, the offspring produced by uniform crossover of x and y is at distance at least n^ϵ from any string in H.*

Proof. Let $z \in H \subseteq \{0,1\}^{n/2}$ be an arbitrary length $n/2$ Hadamard code. Let $B_r(z) \subseteq \{0,1\}^{n/2}$ denote the ball of radius $r < n/4$ around z. By the properties of the dimension $k-1$ Hadamard code, each codeword has minimum distance $2^{k-1}/2$ to any other codeword and thus $d(x,y) \geq 2^{k-1}/2 = n/4$. Therefore, every element of $B_r(z)$ must lie at distance at least $\max\{d(x,z), d(y,z)\} - r \geq n/4 - r$ from at least one of x or y. Therefore, the probability that crossover produces an offspring $w \in B_r(z)$ is at most $(1/2)^{\max\{d(x,w),d(y,w)\}} \leq 2^{-n/4+r}$.

We now bound the probability of the offspring of x and y belonging to a set of solutions that lies within some radius-r ball of any narrow-path solution in $\mathcal{N}$. Let $w \in \{0,1\}^{n/2}$ be the offspring produced by uniform crossover of x and y. There are $|H| = n/2$ distinct Hadamard codes, and $|B_r(z)| \leq (n/2+1)^r$. Applying a union bound, the probability that w lies within a ball of radius r around any Hadamard code $z \in H$ is at most

$$\Pr\left(w \in \bigcup_{z\in H} B_r(z)\right) \leq |H|(n/2+1)^r 2^{-n/4+r} = 2^{-n/4+O(r\log n)}.$$

Setting $r := n^\epsilon$ completes the proof. □

Lemma 1 implies that crossing over two solutions on distinct narrow paths is usually fatal: with overwhelming probability, the offspring is non-viable (or it lies on the wide path) because its narrow-path segment lies sufficiently far from any Hadamard code. Moreover, it is exponentially unlikely that a subsequent mutation operation applied to the offspring could repair the damage, since it would need to flip at least n^{ϵ} bits to get to the nearest narrow path solution.

3 Formal Analysis

We now prove that a high recombination rate favors wide-path solutions during evolution, whereas a low recombination rate favors narrow-path solutions. Individuals that lie on the wide path are *robust* in the following sense. Let x be on the wide path and let y be on a narrow path. Consider any perturbation process that that changes some bits in a string, subject to the constraint that a constant number of bits change in expectation, and that number is concentrated around its expectation (e.g., uniform mutation with a $\Theta(1/n)$ mutation rate, or changing $\Theta(1)$ bits uniformly at random). If x undergoes this perturbation, it is non-viable only with probability $\Theta(k/n) = o(1)$. On the other hand, such a mutation on y results in a non-viable solution already with constant probability. It is therefore easy to see that in a dynamic environment where perturbations occur during evolution (as with our example in Sect. 1.1), a process following the wide path will in general be more successful. The result is also interesting in a static context where the algorithm produces a string robust to changes *after* evolution, or in homologous landscapes in which the optimal solution lies only at the end of the wide path.

3.1 Algorithm

We study a simple population-based evolutionary algorithm equipped with a recombination rate parameter $p_r \in [0, 1]$ that dictates the frequency with which recombination is employed to generate offspring. The $(\mu + 1)$ GA (see Algorithm 1) is a steady-state genetic algorithm that maintains a population of μ elements of $\{0, 1\}^n$ and uses uniform parent selection and truncation survival selection. In each iteration, with probability p_r two parents are chosen uniformly at random without replacement (this condition is not necessary for the result, but necessary for a simpler proof). An offspring is then produced by uniform crossover followed by mutation. Otherwise, with probability $1-p_r$ a single individual is chosen uniformly at random and an offspring is produced by mutation only. We examine its behavior at extremal recombination rates $p_r \in \{0, 1\}$.

We construct the initial population P_0 by selecting exactly one element uniformly at random from each path, hence $\mu = n/2+1$. For each length-$n/2$ string $z' \in H$, we construct a $x \in \{0, 1\}^n$ where the $[x]_1$ is drawn uniformly at random from the set of length-k binary strings with at least $\log k$ ones, $[x]_2 := z'$, and the remaining positions are initialized uniformly at random. For the wide path, we choose a solution uniformly at random from $\mathcal{W}$ by creating a string with the

Algorithm 1. The $(\mu+1)$ GA with recombination rate p_r

```
for $t \leftarrow 0$ to $\infty$ do
    Select $r$ uniformly at random from $[0,1]$;
    Select $\{x,y\} \subseteq P_t$ uniformly at random;
    if $r < p_r$ then $x \leftarrow$ UniformCrossover($x$,$y$);
    Create $z$ by flipping each bit of $x$ independently with probability $1/n$;
    Let $u \in P_t \cup \{z\}$ chosen s.t. $\forall v \in P_t \cup \{z\} \colon f(u) \leq f(v)$;
    $P_{t+1} \leftarrow (P_t \cup \{z\}) \setminus \{u\}$;
```

first k bits set to zero, and the remaining string initialized uniformly at random. Our use of dimension-$\Omega(\log n)$ Hadamard codes requires a linear population size. In this paper we leave the effect of different orders of μ as an open question.

Lemma 2. *Let P_0 be the initial population described above. With probability $1 - e^{-\Omega(n)}$, at least $(1-\epsilon)n/4$ narrow-path solutions have zero fitness.*

Proof. The third segment of each string in P_0 is drawn uniformly at random and so the number of leading ones is geometrically distributed. The event that a string has zero fitness occurs independently with probability 1/2. The count of zero-fitness narrow-path strings in P_0 is binomially distributed and a simple application of Chernoff bounds completes the proof. □

Lemma 3. *Let P_0 be the initial population described above. Let $c > 1$ be the multiplicative constant defined in Eq. (1) and let $a > 1$ be an arbitrary constant. With probability $1 - O(n^{-(a-1)})$, $\max\{f(x) \colon x \in P_0\} \leq ac \log n$.*

Proof. Since the initial fitnesses are geometrically distributed, the probability that a given leading-ones segment has ℓ leading ones is $(1/2)^{\ell+1}$. Taking a union bound over all $\mu = n/2 + 1$ solutions, the probability that no string has more than ℓ leading ones is at least $1 - (n/2+1)(1/2)^{\ell+1}$.

The claim is proved by setting $\ell = a \log n$, since the fitness can be no higher than c multiplied by the number of leading ones in the third segment. □

3.2 No Recombination

We show that mutation-only strategies favor the non-robust narrow-path solutions. We begin by proving it is unlikely that the initial fitness of any wide-path solution is improved within $O(n \log^{1/c} n)$ generations.

Lemma 4. *Let P_0 be the initial population described above and $c > 1$ be the multiplicative constant defined in Eq. (1). Let f_0 be the initial fitness of the wide-path solution in P_0. Then with probability $1 - o(1)$, after $an \log^{1/c} n$ steps of the $(\mu + 1)$ GA with $p_r = 0$, for any constant $a > 1$, every wide path solution has fitness at most f_0.*

Proof. Let $\mathcal{E}$ be the event that the number of wide-path solutions after $an \log^{1/c} n$ iterations is strictly less than $n/\log^2 n$. We first bound the probability of $\mathcal{E}$.

Denote as T_i the waiting time until the number of wide-path solutions increases, measured from the first generation in which there are i wide-path solutions. To jump to the wide-path from a narrow-path, mutation must flip $\log\log n$ bits. This occurs in any generation with probability that vanishes superpolynomially fast, so we assume it does not happen during $an \log^{1/c} n$ steps.

Each T_i is geometrically distributed with success probability at most i/μ since at the very least we must select a wide-path solution for mutation. Let T be first time there are $n/\log^2 n$ wide-path solutions in the population. Thus, $\mathrm{E}(T) = \sum_{i=1}^{n/\log^2 n} \mathrm{E}(T_i) \geq \mu \sum_{i=1}^{n/\log^2 n} 1/i = \Theta(n \log n)$. The probability that the number of wide-path solutions exceeds $n/log^2 n$ after $an \log^{1/c} n$ generations is

$$\Pr(\mathcal{E}) = 1 - \Pr(T \leq an \log^{1/c} n) \geq 1 - \Pr\left(T \leq \frac{a}{\log^{1-1/c} n} E(T)\right) = 1 - o(1).$$

Here we have applied tail bounds on the sum of independent geometric random variables [4]. Assume there are i wide-path solutions in iteration $t < an \log^{1/c} n$. Then, under condition $\mathcal{E}$, the probability that the fitness of *any* wide-path solution is increased is at most $i/(\mu n) \leq 2/(n \log^2 n)$.

Let $\mathcal{F}$ be the event that no wide-path fitness ever increases during $an \log^{1/c} n$ generations. By the law of total probability, $\Pr(\mathcal{F}) \geq \Pr(\mathcal{E}) \Pr(\mathcal{F} \mid \mathcal{E})$, so

$$\Pr(\mathcal{F}) \geq \Pr(\mathcal{E}) \left(1 - \frac{2}{n \log^2 n}\right)^{an \log^{1/c} n} \geq \Pr(\mathcal{E}) \left(1 - \frac{2a}{\log^{2-1/c} n}\right) = 1 - o(1),$$

where we have applied Bernoulli's inequality. □

Theorem 5. *Consider a run of the (μ + 1) GA with $p_r = 0$ initialized with P_0. With probability $1 - o(1)$, there exists a polynomial $\mathrm{poly}(n)$ such that for all $t > \mathrm{poly}(n)$ all elements of P_t are on some narrow path.*

Proof. We first argue that the fitness of the initial wide-path solution is $f_0 \leq \log\log n$ with probability $1 - o(1)$. The fitness of this individual depends only on what position in the leading-ones segment the first zero appears. This value is distributed geometrically with success probability 1/2. So the probability that the first zero appears beyond the ($\log\log n$)-th position is $1 - 2^{-\log\log n} = 1 - o(1)$.

We say a solution $x \in \{0,1\}^n$ is *high-fitness* if $f(x) > f_0$. We now argue that there are many high-fitness narrow-path solutions in P_0. Each narrow-path solution is high-fitness if it has more than $(1/c) \log\log n$ leading ones, because its fitness is then strictly greater than $\log\log n = f_0$. Hence, the probability that an individual narrow-path solution is high-fitness is $2^{-(1/c)\log\log n}$. The initial $n/2$ fitness-values are independent, so by Chernoff bounds, for some positive constant $\gamma > 0$, there are at least $n/(\gamma \log^{1/c} n)$ high-fitness solutions with high probability. For the remainder of the proof, we assume this property holds.

Since an individual in the population can only be replaced by an offspring with a larger or equal fitness value, the count of high-fitness solutions never decreases. Let X_t denote the 0–1 random variable such that $X_t = 1$ if and only if a high-fitness solution is cloned in generation t. We have

$$\Pr(X_t = 1) \geq \frac{1}{\mu}\frac{n}{\gamma \log^{1/c} n}\left(1 - \frac{1}{n}\right)^n \geq \frac{1}{a \log^{1/c} n},$$

for $a > 0$ a positive constant. Let $S = \sum_{t=0}^{an \log^{1/c} n} X_t$. Obviously, S is a lower bound on the number of high-fitness solutions in the population after $an \log^{1/c} n$ generations. Thus we have $\mathrm{E}(S) \geq n = 2(\mu - 1)$ and by Chernoff bounds, $\Pr(S < \mu) \leq e^{-\Omega(n)}$. Hence, with high probability, there are only high-fitness solutions in the population by generation $an \log^{1/c} n$.

Finally, we can apply Lemma 4 to conclude that with probability $1 - o(1)$, no wide-path solution was ever improved during the take-over period of high-fitness solutions. Under these events, no wide-path solution remains in the population.

After this point, a new wide-path solution appears only if a narrow-path solution is mutated onto the wide path. This requires changing at least $\log\log n$ bits in the first segment for which we derive a superpolynomial waiting time w.h.p. When $p_r = 0$, the algorithm is identical to the (μ+1) EA, which solves leading ones in polynomial time [12]. Iterations with no viable offspring only slow the process by a constant factor. After poly(n) steps, all individuals have fitness at least $(n/2 - k) + 1$ and so no wide-path solution will ever be accepted. □

3.3 Full Recombination

We now prove that if the recombination rate is one, the $(\mu + 1)$ GA favors robust wide-path solutions. The following lemma states narrow-path solutions are difficult to create by the crossover operation.

Lemma 6. *Consider a run of the (μ + 1) GA with $p_r = 1$ initialized as above. With probability $1 - o(1)$, no new narrow-path offspring are accepted within n^3 generations.*

Proof. Let $\mathcal{E}_t$ denote the event that the first narrow-path offspring is generated in generation t. We argue that $\Pr(\mathcal{E}_t)$ is sufficiently close to zero for all $t \leq n^3$. Consider a generation in which no new narrow-path offspring have been created yet. There are three possibilities for parent selection: (1) two wide-path solutions are selected as parents, (2) two narrow-path solutions are selected as parents, and (3) wide-path solution and a narrow path solution are selected as parents.

In the first case, the result of uniform crossover must lie on the wide-path since the offspring inherits the entire first segment from both parents. In this case it is up to mutation alone to move the offspring to a narrow path. However, since each narrow-path solution must have $\log\log n$ ones in the first segment, mutation must flip $\log\log n$ bits, which only happens with probability $o(1)$.

In the second case, since we assume no new narrow-path solutions have been produced by generation t, each pair of narrow-path solutions lie on distinct paths. By Lemma 1, crossover between any two narrow-path solutions results in an offspring whose second segment is at least n^ϵ-far from any Hadamard code.

For the third case, we can focus on a single Hadamard code, namely $h_0 = 0^{n/2}$, and then take a union bound over all $n/2$ codes. We define the *Hamming weight* of a binary string to be the number of ones it contains. Let X_t be the minimum Hamming weight in the second segment over all wide-path solutions at time t. If a wide-path solution is recombined with the unique narrow-path solution $y \in \mathcal{N}$ with $[y]_2 = h_0$, then the expected Hamming weight of the offspring is $X_t/2$. Otherwise, if we cross a wide-path solution with some string z having $m > 0$ ones in $[z]_2$, the expected Hamming weight of the offspring is $X_t/2 + m/2$. The probability of selecting y is $1/\mu$. It is straight-forward to apply a negative drift argument [8,9] on the potential $\log(X_t)$ to show X_t does not hit zero within n^3 generations with probability at most $1/n^{4+\delta}$ for a constant $\delta > 0$. In total, the probability for the third case is at most $2/n^{3+\delta}$.

Therefore, letting $p = \max_{0 \le t \le n^3}\{\Pr(\mathcal{E}_t)\}$, the probability that the count of narrow-path solutions does not increase in the first n^3 steps is bounded by $\prod_{t=1}^{n^3} (1 - \Pr(\mathcal{E}_t)) \geq (1-p)^{n^3} \geq 1 - pn^3 = 1 - o(1)$. □

Theorem 7. *Starting from the initial population described above, with high probability, after $T = O(n^2 \log n)$ iterations of the $(\mu + 1)$ GA, all elements of P_T are on the wide path and remain there for any polynomial number of steps.*

Proof. We show that with probability $1 - o(1)$ the entire population converges to the wide path in $O(n^2 \log n)$ steps. Subsequently, only wide-path solutions reproduce so we get superpolynomial waiting time for new narrow-path solutions.

Define W_t and Z_t to be the count of wide-path solutions and zero-fitness narrow path solutions in P_t, respectively. We condition on the following set of events, each holding with high probability: (1) a narrow-path offspring does not appear within n^3 steps (Lemma 6), (2) $Z_0 \geq (1-\epsilon)n/4$ (Lemma 2), and (3) the fitness of any solution in P_0 is at most $ac \log n$ for constants $a, c > 1$ (Lemma 3).

We divide a run of the $(\mu + 1)$ GA into two phases. The first phase begins at $t = 0$ and lasts until there are $\Omega(n)$ wide-path solutions in the population. Let $T_1 = \inf\{t \in \mathbb{N} \colon W_t > \mu/8\}$. The first phase begins at $t = 0$ and ends at $t = T_1$. During this phase, since we assume no narrow-path solutions are spontaneously created, $W_t \leq \mu/8$ and $Z_t \geq Z_0 - \mu/8$. Moreover, $W_{t+1} - W_t \geq 0$ for all $t \leq T_1$.

During this phase, a wide-path solution is chosen as a parent with probability W_t/μ and the resulting offspring is on the wide path with probability at least $1/(2en)$. Under this event, $W_{t+1} = W_t + 1$ only if a narrow-path solution is replaced in the selection phase. Since the fitness of the offspring is at least zero, the probability that a narrow path solution is selected for deletion is at least $Z_t/\mu \geq (3 - 4\epsilon)/8$. Thus at each iteration $t \leq T_1$ in the first phase, we have $\mathrm{E}(W_{t+1} - W_t \mid W_t) \geq W_t(3 - 4\epsilon)/(16\mu en) = \Omega\big(W_t/n^2\big)$. We can bound the hitting time of W_t to $\mu/8 = n/16$ using the General Drift Theorem of Lehre and Witt [6] to get $T_1 = O(n^2 \log n)$ with probability $1 - o(1)$.

The second phase begins at time $T_1 + 1$. In this phase, $W_t \geq \mu/8$ so two wide-path solutions are selected as parents for crossover with probability $\Omega(1)$.

If two wide-path solutions are selected as parents, the result of crossover (before mutation) must be on the wide path, and must have a fitness at least as high as the lowest fitness of the parents. With probability at least $1/(en)$, mutation can then flip the first zero in the leading-ones segment to improve the fitness of the offspring by at least 1. This offspring is accepted since it is strictly more fit than at least one element of P_t (its least fit parent). We call such a result a *success*.

The probability of a success in each iteration is $\Omega(1/n)$ and independent. After $c'\mu n$ iterations for an appropriate constant $c' > 0$, we have had at least μ successes in expectation. We call a sequence of μ successes a *round*. Let $m = \min\{f(x) \colon x \in P_{T_1+1}\} \geq 0$ be the minimum fitness in the population at the start of phase two. After one round, μ offspring have been accepted with fitness at least $m+1$ so the minimum fitness in the population after the first round is at least 1. After $1 + ac\log n$ rounds, the minimum fitness is at least $(m+1) + ac\log n > ac\log n$. Since no new narrow-path solutions spontaneously appear during this time, it follows by Lemma 3 that all narrow-path solutions present in the initial population must have been replaced by wide-path solutions during this phase.

Applying Chernoff bounds to the success count, with probability $1 - e^{-\Omega(n)}$, each round takes at most $(1+\epsilon)c'\mu n = O(n^2)$ steps and we conclude all narrow-path solutions are replaced after $O(n^2 \log n)$ rounds during the second phase. □

Acknowledgements. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 618091 (SAGE) and the German Research Foundation (DFG) under grant agreement no. FR 2988 (TOSU).

References

1. Arora, S., Barak, B.: Computational Complexity: A Modern Approach, 1st edn. Cambridge University Press, New York (2009)
2. Doerr, B., Happ, E., Klein, C.: Crossover can provably be useful in evolutionary computation. Theor. Comput. Sci. **425**, 17–33 (2012)
3. Gardner, A., Kalinka, A.T.: Recombination and the evolution of mutational robustness. J. Theor. Biol. **241**(4), 707–715 (2006)
4. Janson, S.: Tail bounds for sums of geometric and exponential variables. (2014). http://www2.math.uu.se/svante/papers/sjN14.pdf
5. Kötzing, T., Sudholt, D., Theile, M.: How crossover helps in pseudo-Boolean optimization. In: GECCO, pp. 989–996 (2011)
6. Lehre, P.K., Witt, C.: General drift analysis with tail bounds. arXiv:1307.2559 [cs.NE] (2013)
7. Lehre, P.K., Yao, X.: Crossover can be constructive when computing unique input-output sequences. Soft Comput. **15**(9), 1675–1687 (2011)
8. Oliveto, P.S., Witt, C.: Simplified drift analysis for proving lower bounds in evolutionary computation. Algorithmica **59**(3), 369–386 (2011)
9. Oliveto, P.S., Witt, C.: Erratum: simplified drift analysis for proving lower bounds in evolutionary computation. arXiv:1211.7184 [cs.NE] (2012)

10. Sudholt, D.: Crossover is provably essential for the Ising model on trees. In: GECCO, pp. 1161–1167 (2005)
11. Sudholt, D.: Crossover speeds up building-block assembly. In: GECCO, pp. 689–696 (2012)
12. Witt, C.: Runtime analysis of the $(\mu + 1)$ EA on simple Pseudo-Boolean functions. Evol. Comput. **14**, 65–86 (2006)
13. Yao, X.: Evolving artificial neural networks. Proc. IEEE **87**(9), 1423–1447 (1999)