

The Benefit of Recombination in Noisy Evolutionary Search

Tobias Friedrich, Timo Kötzing^(✉), Martin S. Krejca,
and Andrew M. Sutton

Hasso Plattner Institute, Potsdam, Germany
timo.koetzing@hpi.de

Abstract. Practical optimization problems frequently include uncertainty about the quality measure, for example due to noisy evaluations. Thus, they do not allow for a straightforward application of traditional optimization techniques. In these settings meta-heuristics are a popular choice for deriving good optimization algorithms, most notably evolutionary algorithms which mimic evolution in nature. Empirical evidence suggests that genetic recombination is useful in uncertain environments because it can stabilize a noisy fitness signal. With this paper we want to support this claim with mathematical rigor.

The setting we consider is that of noisy optimization. We study a simple noisy fitness function that is derived by adding Gaussian noise to a monotone function. First, we show that a classical evolutionary algorithm that does not employ sexual recombination (the $(\mu+1)$ -EA) cannot handle the noise efficiently, regardless of the population size. Then we show that an evolutionary algorithm which does employ sexual recombination (the Compact Genetic Algorithm, short: cGA) can handle the noise using a graceful scaling of the population.

1 Introduction

Heuristic optimization is widely used in practice for solving hard optimization problems for which no efficient problem-specific algorithm is known. Such problems are typically very large, noisy and constrained and cannot be solved by simple textbook algorithms. The inspiration for heuristic general-purpose problem solvers often comes from nature. A well-known example is simulated annealing, which is inspired from physical annealing in metallurgy. The largest and probably most successful class, however, are biologically-inspired algorithms, especially *evolutionary algorithms*.

Evolutionary and Genetic Algorithms. Evolutionary Algorithms (EAs) were introduced in the 1960s and have been successfully applied to a wide range of complex engineering and combinatorial problems [1, 10, 24]. Like Darwinian evolution in nature, evolutionary algorithms construct new solutions from old ones and select the fitter ones to continue to the next iteration. The construction of new solutions from old ones, so-called reproduction, can

be *asexual* (mutation of a single individual) or *sexual* (crossover of several individuals). An EA that uses sexual reproduction is typically called *Genetic Algorithm* (GA). Since the beginning of EAs, it has been argued that GAs should be more powerful than pure EAs, which use only asexual reproduction [13]. This was debated for decades, but theoretical results and explanations on crossover are still scarce. There are some results for simple artificial test functions, where it was proven that a GA asymptotically outperforms an EA without crossover [16, 17, 20, 25, 31, 35] and the other way around [30]. However, these artificial test functions are typically tailored to the specific algorithm and proof technique and the results give little insight into the advantage of sexual reproduction on realistic problems. There are also a few theoretical results for problem-specific algorithms and representations, namely coloring problems inspired by the Ising model [32] and the all-pairs shortest path problem [5]. For a nice overview of different aspects where populations and sexual recombination are beneficial for optimization of static fitness functions, see [29].

The underlying search space of many optimization problems is the set $\{0, 1\}^n$ of all length- n bit strings. Many problems (including combinatorial ones such as the minimum spanning tree problem) have a straightforward formulation as an optimization problem on $\{0, 1\}^n$. Many evolutionary algorithms are applicable to this search space without further modification adaption, and most formal analyses of evolutionary algorithms consider this search space. A popular simple fitness function on this search space is *OneMax*, which uses the number of 1s in a bit string as fitness value. A cornerstone of the analysis of any search heuristic is an analysis of its performance on the OneMax function [8, 37], and studying the class of OneMax functions has also lead to several breakthroughs in the field of black-box complexity [4, 9]. Finally, there are also works analyzing the use of crossover for the OneMax function [7, 28, 33].

Noisy Search. Heuristic optimization methods are typically not used for simple problems, but for rather difficult problems in *uncertain environments*. Evolutionary algorithms are very popular in settings including uncertainties; see [2] for a survey on examples in combinatorial optimization, but also [18] for an excellent survey also discussing different sources of uncertainty. Uncertainty can be modeled by a *probabilistic* fitness function, that is, a search point can have different fitness values each time it is evaluated. One way to deal with this is to replace fitness evaluations with an average of a (large) sample of fitness evaluations and then proceed as if there was no noise. In this work we show that generic GAs (with sexual reproduction) can overcome noise much more efficiently than using this naive approach. To do this in a rigorous manner, we assume *additive posterior noise*, that is, each time the fitness value of a search point is evaluated, we add a noise value drawn from some distribution. This model was studied in evolutionary algorithms without crossover in [6, 11, 12, 14, 34].

We will consider centered Gaussian noise with variance σ^2 and use OneMax as the underlying fitness function. Already such a seemingly simple setting poses difficulties to the analysis of evolutionary algorithms, as these algorithms are

not developed with the analysis in mind. Particularly algorithms with sexual recombination have been resisting a mathematical analysis.

Our Results. We are interested in studying how well search heuristics can cope with noise, for which we use the concept of *graceful scaling* (Definition 2); intuitively, a search heuristic scales gracefully with noise if (polynomially) more noise can be compensated by (polynomially) more resources.

We first prove a sufficient condition for when a noise model is intractable for optimization by the classical $(\mu + 1)$ -EA (Theorem 4) and show that this implies that this simple asexual algorithm does not scale gracefully for Gaussian noise (Corollary 5). On the other hand, we study the compact GA (cGA), which models a genetic algorithm, and show how its gene-pool recombination operator is able to “smooth” the noise sufficiently to exhibit graceful scaling (Theorem 9).

We proceed in Sect. 2 by formalizing our setting and introducing the algorithms we consider. In Sect. 3 we give our results. Note that in this extended abstract, we omit many proof details and provide only proof sketches due to space constraints. We conclude the paper in Sect. 4.

2 Preliminaries

In the remainder of the paper, we will study a particular function class (OneMax) and a particular noise distribution (Gaussian, parametrized by the variance). Let $\sigma^2 \geq 0$. We define the noisy OneMax function $\text{OM}_{[\sigma^2]}: \{0, 1\}^n \rightarrow \mathbb{R} := x \mapsto \|x\|_1 + Z$ where $\|x\|_1 := |\{i: x_i = 1\}|$ and Z is a normally distributed random variable $Z \sim \mathcal{N}(0, \sigma^2)$ with zero mean and variance σ^2 .

The following proposition gives tail bounds for Z by using standard estimates of the complementary error function [36].

Proposition 1. *Let Z be a zero-mean Gaussian random variable with variance σ^2 . For all $t > 0$ we have*

$$\Pr(Z < -t) = \frac{1}{2} \operatorname{erfc}\left(\frac{t}{\sigma\sqrt{2}}\right) \leq \frac{1}{2} e^{-t^2/(2\sigma^2)}$$

and asymptotically for large $t > 0$,

$$\Pr(Z < -t) = \frac{1}{1 + o(1)} \frac{\sigma}{\sqrt{2\pi}t} e^{-t^2/(2\sigma^2)}.$$

2.1 Algorithms

The $(\mu + 1)$ -EA, defined in Algorithm 1, is a simple mutation-only evolutionary algorithm that maintains a population of μ solutions and uses elitist survival selection. It derives its name from maintaining a population of μ individuals (randomly initialized) and generating one new individual each iteration by mutating a parent chosen uniformly at random from the current population. Then it evaluates the fitness of all individuals and chooses one with minimal value to be

Algorithm 1. The $(\mu + 1)$ -EA

```

1  $t \leftarrow 0$ ;
2  $P_t \leftarrow \mu$  elements of  $\{0, 1\}^n$  u.a.r.;
3 while termination criterion not met do
4   Select  $x \in P_t$  u.a.r.;
5   Create  $y$  by flipping each bit of  $x$  independently with probability  $1/n$ ;
6   Let  $z \in P_t \cup \{y\}$  chosen s.t.  $\forall v \in P_t \cup \{y\}: f(z) \leq f(v)$ ;
7    $P_{t+1} \leftarrow P_t \cup \{y\} \setminus \{z\}$ ;
8    $t \leftarrow t + 1$ ;

```

removed from the population, so that again μ individuals proceed to the next generation.

The compact genetic algorithm (cGA) [15] is a genetic algorithm that maintains a population of size K *implicitly* in memory. Rather than storing each individual separately, the cGA only keeps track of population *allele frequencies* and updates these frequencies during evolution. Offspring are generated according to these allele frequencies, which is similar to what occurs in models of sexually-recombining natural populations. Indeed, the offspring generation procedure can be viewed as so-called *gene pool recombination* introduced by Mühlenbein and Paaß [23] in which all K members participate in uniform recombination. Since the cGA evolves a probability distribution, it is also a type of *estimation of distribution algorithm* (EDA). The correspondence between EDAs and models of sexually recombining populations has already been noted [22], and Harik et al. [15] demonstrate empirically that the behavior of the cGA is equivalent to a simple genetic algorithm at least on simple problems.

Algorithm 2. The compact GA

```

1  $t \leftarrow 0$ ;
2  $p_{1,t} \leftarrow p_{2,t} \leftarrow \dots \leftarrow p_{n,t} \leftarrow 1/2$ ;
3 while termination criterion not met do
4   for  $i \in \{1, \dots, n\}$  do
5      $x_i \leftarrow 1$  with probability  $p_{i,t}$ ,  $x_i \leftarrow 0$  with probability  $1 - p_{i,t}$ ;
6   for  $i \in \{1, \dots, n\}$  do
7      $y_i \leftarrow 1$  with probability  $p_{i,t}$ ,  $y_i \leftarrow 0$  with probability  $1 - p_{i,t}$ ;
8   if  $f(x) < f(y)$  then swap  $x$  and  $y$ ;
9   for  $i \in \{1, \dots, n\}$  do
10    if  $x_i > y_i$  then  $p_{i,t+1} \leftarrow p_{i,t} + 1/K$ ;
11    if  $x_i < y_i$  then  $p_{i,t+1} \leftarrow p_{i,t} - 1/K$ ;
12    if  $x_i = y_i$  then  $p_{i,t+1} \leftarrow p_{i,t}$ ;
13    $t \leftarrow t + 1$ ;

```

The first rigorous analysis of the cGA is due to Droste [8] who gave a general runtime lower bound for all pseudo-Boolean functions, and a general upper

bound for all linear pseudo-Boolean functions. Defined in Algorithm 2, the cGA maintains for all times $t \in \mathbb{N}_0$ a frequency vector $(p_{1,t}, p_{2,t}, \dots, p_{n,t}) \in [0, 1]^n$. In the t -th iteration, two strings x and y are sampled independently from this distribution where $\Pr(x = z) = \Pr(y = z) = \left(\prod_{i: z_i=1} p_{i,t}\right) \times \left(\prod_{i: z_i=0} (1 - p_{i,t})\right)$ for all $z \in \{0, 1\}^n$. The cGA then compares the objective values of x and y , and updates the distribution by advancing $p_{i,t}$ toward the component of the winning string by an additive term. This small change in allele frequencies is equivalent to a population undergoing steady-state binary tournament selection [15].

Let F be a family of pseudo-Boolean functions $(F_n)_{n \in \mathbb{N}}$ where each F_n is a set of functions $f: \{0, 1\}^n \rightarrow \mathbb{R}$. Let D be a family of distributions $(D_v)_{v \in \mathbb{R}}$ such that for all $D_v \in D$, $\mathbb{E}(D_v) = 0$. We define F with additive posterior D -noise as the set $F[D] := \{f_n + D_v: f_n \in F_n, D_v \in D\}$.

Definition 2. *An algorithm A scales gracefully with noise on $F[D]$ if there is a polynomial q such that, for all $g_{n,v} = f_n + D_v \in F[D]$, there exists a parameter setting p such that $A(p)$ finds the optimum of f_n using at most $q(n, v)$ calls to $g_{n,v}$.*

Algorithms that operate in the presence of noise often depend on *a priori* knowledge of the noise intensity (measured by the variance). In such cases, the following scheme can always be used to transform such algorithms into one that has no knowledge of the noise character. Suppose $A(\sigma^2)$ is an algorithm that solves a noisy function with variance at most σ^2 within $T_\delta(\sigma^2)$ steps with probability at least $1 - \delta$. A *noise-oblivious scheme* for A is in Algorithm 3.

Algorithm 3. Noise-oblivious scheme for A

```

1  $i \leftarrow 0$ ;
2 repeat until solution found
3   Run  $A(2^i)$  for  $T_\delta(2^i)$  steps;
4    $i \leftarrow i + 1$ ;
```

If an algorithm A scales gracefully with noise, then the noise oblivious scheme for A scales gracefully with noise. The following proposition holds by a simple inductive argument.

Proposition 3. *Suppose $f_{n,v} \in F[D]$ is a noisy function with unknown variance v . Fixing n and assume that, for all $c > 0$ and all x , $cT_\delta(x) \leq T_\delta(cx)$. Then for any $s \in \mathbb{Z}^+$, the noise-oblivious scheme optimizes $f_{n,v}$ in at most $T_\delta(2^s v)$ steps with probability at least $1 - \delta^s$.*

3 Results

We derive rigorous bounds on the optimization time, defined as the first hitting time of the process to the true optimal solution (1^n) of $\text{OM}_{[\sigma^2]}$, on a mutation-based approach and the compact genetic algorithm.

3.1 Mutation-Based Approach

In this section we consider the $(\mu + 1)$ -EA. We will first, in Theorem 4, give a sufficient condition for when a noise model is intractable for optimization by a $(\mu + 1)$ -EA. While uniform selection removes any individual from the population with probability $1/(\mu + 1)$, the condition of Theorem 4 requires that the noise is strong enough so that the $(\mu + 1)$ -EA will remove any individual with at least half that probability. Then we will show that, in the case of additive posterior noise sampled from a Gaussian distribution, this condition is fulfilled if the noise is large enough, showing that the $(\mu + 1)$ -EA cannot deal with arbitrary Gaussian noise (see Corollary 5).

Theorem 4. *Let $\mu \geq 1$ and D a distribution on \mathbb{R} . Let Y be the random variable describing the minimum over μ independent copies of D . Suppose*

$$\Pr(Y > D + n) \geq \frac{1}{2(\mu + 1)}.$$

Consider optimization of OneMax with reevaluated additive posterior noise from D by $(\mu + 1)$ -EA. Then, for μ bounded from above by a polynomial, the optimum will not be evaluated after polynomially many iterations w.h.p.

Proof Sketch. For all t and all $i \leq n$ let X_i^t be the random variable describing the proportion of individuals in the population of iteration t with exactly i 1s. The proof is by induction on t that

$$\forall t, \forall i \geq an: \mathbb{E}(X_i^t) \leq b^{an-i},$$

where a, b and c are specifically chosen constants. In other words, the expected number of individuals with i 1s is decaying exponentially with i after an . This will give the desired result with a simple union bound over polynomially many time steps. \square

We apply Theorem 4 to show that large noise levels make it impossible for the $(\mu + 1)$ -EA to efficiently optimize when the noise is significantly larger than the range of objective values. The proof is a simple exercise in bounding the tails of a Gaussian distribution using Proposition 1.

Corollary 5. *Consider optimization of $\text{OM}_{[\sigma^2]}$ by $(\mu + 1)$ -EA. Suppose $\sigma^2 \geq n^3$ and μ bounded from above by a polynomial in n . Then the optimum will not be evaluated after polynomially many iterations w.h.p.*

3.2 Compact GA

Let T^* be the optimization time of the cGA on $\text{OM}_{[\sigma^2]}$, namely, the first time that it generates the underlying “true” optimal solution 1^n . We consider the stochastic process $X_t = n - \sum_{i=1}^n p_{i,t}$ and bound the optimization time by $T = \inf\{t \in \mathbb{N}_0: X_t = 0\}$. Clearly $T^* \leq T$ since the cGA produces 1^n in the T -th iteration almost surely. However, T^* and T can be infinite when there is

a $t < T^*$ where $p_{i,t} = 0$ since the process can never subsequently generate any string x with $x_i = 1$. To circumvent this, Droste [8] estimates $E(T^*)$ conditioned on the event that T^* is finite, and then bounds the probability of finite T^* . In this paper, we will prove that as long as K is large enough, the optimization time is finite (indeed, polynomial) with high probability. To prove our result, we need the following drift theorem.

Theorem 6 (Tail Bounds for Multiplicative Drift [3, 21]). *Let $\{X_t : t \in \mathbb{N}_0\}$ be a sequence of random variables over a set $S \subseteq \{0\} \cup [x_{\min}, x_{\max}]$ where $x_{\min} > 0$. Let T be the random variable that denotes the earliest point in time $t \geq 0$ such that $X_t = 0$. If there exists $0 < \delta < 1$ such that $E(X_t - X_{t+1} \mid T > t, X_t) \geq \delta X_t$, then*

$$\Pr\left(T > \frac{\lambda + \ln(X_0/x_{\min})}{\delta} \mid X_0\right) \leq e^{-\lambda} \text{ for all } \lambda > 0.$$

The following lemma bounds the drift on X_t , conditioned on the event that no allele frequency gets too small.

Lemma 7. *Consider the cGA optimizing $\text{OM}_{[\sigma^2]}$ and let X_t be the stochastic process defined above. Assume that there exists a constant $a > 0$ such that $p_{i,t} \geq a$ for all $i \in \{1, \dots, n\}$ and that $X_t > 0$, then $E(X_t - X_{t+1} \mid X_t) \geq \delta X_t$ where $1/\delta = \mathcal{O}(\sigma^2 K \sqrt{n})$.*

Proof Sketch. Let x and y be the offspring generated in iteration t and $Z_t = \|x\|_1 - \|y\|_1$. Then $Z_t = Z_{1,t} + \dots + Z_{n,t}$ where

$$Z_{i,t} = \begin{cases} -1 & \text{if } x_i = 0 \text{ and } y_i = 1, \\ 0 & \text{if } x_i = y_i, \\ 1 & \text{if } x_i = 1 \text{ and } y_i = 0. \end{cases}$$

Let \mathcal{E} denote the event that in line 8, the evaluation of $\text{OM}_{[\sigma^2]}$ correctly ranks x and y . Then

$$E(X_t - X_{t+1} \mid X_t) = \frac{E(|Z_t|)}{K} (1 - 2\Pr(\bar{\mathcal{E}})).$$

Using combinatorial arguments and properties of the Poisson-Binomial distribution, the expectation of $|Z_t|$ can be bounded from below by $aX_t\sqrt{2/n}$. The proof can then be completed by bounding the probability that x and y are incorrectly ranked, which is at most $\frac{1}{2}(1 - \Omega(\sigma^{-2}))$. This follows from straightforward deviation bounds on the normal distribution derived from Proposition 1. \square

To use Lemma 7, we require that the allele frequencies stay large enough during the run of the algorithm. Increasing the effective population size K obviously translates to finer-grained allele frequency values, which means slower dynamics for $p_{i,t}$. Indeed, provided that K is set sufficiently large, the allele frequencies remain above an arbitrary constant for any polynomial number of iterations with very high probability. This is captured by the following lemma.

Lemma 8. *Consider the cGA optimizing $\text{OM}_{[\sigma^2]}$ with $\sigma^2 > 0$. Let $0 < a < 1/2$ be an arbitrary constant and $T' = \min\{t \geq 0: \exists i \in [n], p_{i,t} \leq a\}$. If $K = \omega(\sigma^2 \sqrt{n} \log n)$, then for every polynomial $\text{poly}(n)$, n sufficiently large, $\Pr(T' < \text{poly}(n))$ is superpolynomially small.*

Proof Sketch. Let $i \in [n]$ be an arbitrary index. Let $\{Y_t: t \in \mathbb{N}_0\}$ be the tochastic process $Y_t = (1/2 - p_{i,t}) K$. The proof begins by first showing that

$$\mathbb{E}(Y_t \mid Y_1, \dots, Y_{t-1}) \leq Y_{t-1} - \Omega(\sigma^{-2}) \frac{1}{\sqrt{n}}.$$

The idea behind this claim is a follows. Obviously, $Y_t - Y_{t-1} \in \{-1, 0, 1\}$ and it suffices to bound the conditional expectation of this difference in one step. Again let x and y be the offspring generated in iteration t . The argument proceeds by considering the substrings of x and y induced by the remaining indexes (in $[n] \setminus \{i\}$), which are by definition statistically independent. Let \mathcal{E} denote the event that these substrings are equal. If $x_i \neq y_i$, then whichever string contains a 1 in the i -th position has a strictly greater “true” fitness, and the change in Y_t with respect to Y_{t-1} depends only on the event that $\text{OM}_{[\sigma^2]}$ incorrectly ranks x and y . This probability can be bounded as in Lemma 7 and the $1/\sqrt{n}$ factor comes a bound on $\Pr(\mathcal{E})$ that arises from the fact that the number of positions $j \in [n] \setminus \{i\}$ where $x_j \neq y_j$ has a Poisson-Binomial distribution. It is then straightforward to show that the contributions to the expected difference conditioned on \mathcal{E} remains strictly negative. This is simply an exercise in checking the remaining possibilities and bounding their probability.

The proof is then finished by applying a refinement to the negative drift theorem of Oliveto and Witt [26, 27] (cf. Theorem 3 of [19]). Implicitly ignoring self-loops in the Markov chain (which can only result in a slower process), we have $Y_1 = 0$ and $|Y_t - Y_{t+1}| \leq 1 < \sqrt{2}$, and thus for all $s \geq 0$,

$$\Pr(T' \leq s) \leq s \exp\left(-\frac{(1/2 - a)K|\epsilon|}{32}\right),$$

with $\epsilon = -\Omega(\sigma^{-2}/\sqrt{n})$. Since $K = \omega(\sigma^2 \sqrt{n} \log n)$, $\Pr(T' \leq s) = sn^{-\omega(1)}$.

So, for any polynomial $s = \text{poly}(n)$, with probability superpolynomially close to one, Y_s has not yet reached a state larger than $(1/2 - a)K$, and so $p_{i,t} > a$ for all $0 \leq t \leq s$. As this holds for arbitrary i , applying a union bound retains a superpolynomially small probability that any of the n frequencies have gone below a by $s = \text{poly}(n)$ steps. \square

It is now straightforward to prove that the optimization time of the cGA is polynomial in the problem size and the noise variance. This is in contrast to the mutation-based $(\mu + 1)$ -EA, which fails when the variance becomes large. This means the cGA scales gracefully with noise in the sense of Definition 2 applied to the $\text{OM}_{[\sigma^2]}$ noise model.

Theorem 9. *Consider the cGA optimizing $\text{OM}_{[\sigma^2]}$ with variance $\sigma^2 > 0$. If $K = \omega(\sigma^2 \sqrt{n} \log n)$, then with probability $1 - o(1)$, the cGA finds the optimum after $\mathcal{O}(K\sigma^2 \sqrt{n} \log Kn)$ steps.*

Proof. We will consider the drift of the stochastic process $\{X_t : t \in \mathbb{N}_0\}$ over the state space $S \subseteq \{0\} \cup [x_{\min}, x_{\max}]$ where $X_t = n - \sum_{i=1}^n p_{i,t}$. Hence, $x_{\min} = 1/K$.

Fix a constant $0 < a < 1/2$. We say the process has *failed* by time t if there exists some $s \leq t$ and some $i \in [n]$ such that $p_{i,s} \leq a$. Let $T = \min\{t \in \mathbb{N}_0 : X_t = 0\}$. Assuming the process never fails, by Lemma 7, the drift of $\{X_t : t \in \mathbb{N}_0\}$ in each step is bounded by $E(X_t - X_{t+1} \mid X_t = s) \geq \delta X_t$ where $1/\delta = \mathcal{O}(\sigma^2 K \sqrt{n})$. By Theorem 6, $\Pr(T > (\ln(X_0/x_{\min}) + \lambda)/\delta) \leq e^{-\lambda}$. Choosing $\lambda = d \ln n$ for any constant $d > 0$, the probability that $T = \Omega(K\sigma^2\sqrt{n} \log Kn)$ is at most n^{-d} .

Letting \mathcal{E} be the event that the process has not failed by $\mathcal{O}(K\sigma^2\sqrt{n} \log Kn)$ steps, by the law of total probability, the hitting time of $X_t = 0$ is bounded by $\mathcal{O}(K\sigma^2\sqrt{n} \log Kn)$ with probability $(1 - n^{-d})\Pr(\mathcal{E}) = 1 - o(1)$ where we can apply Lemma 8 to bound the probability of \mathcal{E} . \square

4 Conclusions

In this paper we have examined the benefit of sexual recombination in evolutionary optimization on the fitness function $\text{OM}_{[\sigma^2]}$. The noise-free function ($\text{OM}_{[0]}$) is efficiently optimized by a simple hillclimber in $\Theta(n \log n)$ steps (this well-known statement follows from a coupon collector argument). Corollary 5 asserts that mutation-only (and by extension, simple hillclimbers) cannot optimize $\text{OM}_{[\sigma^2]}$ in polynomial time without using some kind of resampling strategy to reduce the variance. The intuitive reason for this is that the probability of generating and accepting a worse individual becomes larger than the probability of generating and accepting a better individual: mutation has a bias towards bit strings with about as many 0s as 1s, and for high noise the probability of accepting slightly worse individuals is about 1/2. Thus, mutation-only evolutionary algorithms do not scale gracefully in the sense that they cannot optimize noisy functions in polynomial time when the noise intensity is sufficiently high.

On the other hand, we proved that a genetic algorithm that uses gene pool recombination can always optimize noisy OneMax ($\text{OM}_{[\sigma^2]}$) in expected polynomial time, subject only to the condition that the noise variance σ^2 is bounded by some polynomial in n . Intuitively, the cGA can leverage the *sexual* operation of gene pool recombination to average out the noise and follow the underlying objective function signal.

Our results highlight the importance of understanding the influence of different search operators in uncertain environments, and suggest that algorithms such as the compact genetic algorithm that use some kind of recombination are able to scale gracefully with noise.

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).

References

1. Bäck, T., Fogel, D.B., Michalewicz, Z. (eds.): Handbook of Evolutionary Computation, 1st edn. IOP Publishing Ltd., Bristol (1997)
2. Bianchi, L., Dorigo, M., Gambardella, L., Gutjahr, W.: A survey on metaheuristics for stochastic combinatorial optimization. *Nat. Comput.* **8**, 239–287 (2009)
3. Doerr, B., Goldberg, L.A.: Adaptive drift analysis. *Algorithmica* **65**, 224–250 (2013)
4. Doerr, B., Winzen, C.: Playing mastermind with constant-size memory. In: Proceedings of STACS 2012, pp. 441–452 (2012)
5. Doerr, B., Happ, E., Klein, C.: Crossover can provably be useful in evolutionary computation. *Theor. Comput. Sci.* **425**, 17–33 (2012a)
6. Doerr, B., Hota, A., Kötzing, T.: Ants easily solve stochastic shortest path problems. In: Proceedings of GECCO 2012, pp. 17–24 (2012b)
7. Doerr, B., Doerr, C., Ebel, F.: From black-box complexity to designing new genetic algorithms. *Theor. Comput. Sci.* **567**, 87–104 (2015)
8. Droste, S.: A rigorous analysis of the compact genetic algorithm for linear functions. *Nat. Comput.* **5**, 257–283 (2006)
9. Droste, S., Jansen, T., Wegener, I.: Upper and lower bounds for randomized search heuristics in black-box optimization. *Theory Comput. Syst.* **39**, 525–544 (2006)
10. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer, Heidelberg (2003)
11. Feldmann, M., Kötzing, T.: Optimizing expected path lengths with ant colony optimization using fitness proportional update. In: Proceedings of FOGA 2013, pp. 65–74 (2013)
12. Gießen, C., Kötzing, T.: Robustness of populations in stochastic environments. In: Proceedings of GECCO 2014, pp. 1383–1390 (2014)
13. Goldberg, D.E.: Genetic Algorithms in Search Optimization and Machine Learning. Addison-Wesley, Boston (1989)
14. Gutjahr, W., Pflug, G.: Simulated annealing for noisy cost functions. *J. Global Optim.* **8**, 1–13 (1996)
15. Harik, G.R., Lobo, F.G., Goldberg, D.E.: The compact genetic algorithm. *IEEE Trans. Evol. Comp.* **3**, 287–297 (1999)
16. Jansen, T., Wegener, I.: Real royal road functions—where crossover provably is essential. *Discrete Appl. Math.* **149**, 111–125 (2005)
17. Jansen, T., Wegener, I.: The analysis of evolutionary algorithms - a proof that crossover really can help. *Algorithmica* **34**, 47–66 (2002)
18. Jin, Y., Branke, J.: Evolutionary optimization in uncertain environments—a survey. *IEEE Trans. Evol. Comp.* **9**, 303–317 (2005)
19. Kötzing, T.: Concentration of first hitting times under additive drift. In: Proceedings of GECCO 2014, pp. 1391–1397 (2014)
20. Kötzing, T., Sudholt, D., Theile, M.: How crossover helps in pseudo-boolean optimization. In: Proceedings of GECCO 2011, pp. 989–996 (2011)
21. Lehre, P.K., Witt, C.: Concentrated hitting times of randomized search heuristics with variable drift. In: Ahn, H.-K., Shin, C.-S. (eds.) ISAAC 2014. LNCS, vol. 8889, pp. 686–697. Springer, Heidelberg (2014)
22. Mühlenbein, H., Paaß, G.: From recombination of genes to the estimation of distributions I. Binary parameters. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN 1996. LNCS, vol. 1141, pp. 178–187. Springer, Heidelberg (1996)

23. Mühlenbein, H., Voigt, H.-M.: Gene pool recombination in genetic algorithms. In: Osman, I.H., Kelly, J.P. (eds.) *Meta-Heuristics*, pp. 53–62. Springer, New York (1996)
24. Neumann, F., Witt, C.: *Bioinspired Computation in Combinatorial Optimization - Algorithms and Their Computational Complexity*. Natural Computing Series. Springer, Heidelberg (2010)
25. Neumann, F., Oliveto, P.S., Rudolph, G., Sudholt, D.: On the effectiveness of crossover for migration in parallel evolutionary algorithms. In: *Proceedings of GECCO 2011*, pp. 1587–1594 (2011)
26. Oliveto, P.S., Witt, C.: Simplified drift analysis for proving lower bounds in evolutionary computation. *Algorithmica* **59**, 369–386 (2011)
27. Oliveto, P.S., Witt, C.: Erratum: simplified drift analysis for proving lower bounds in evolutionary computation (2012). [arXiv:1211.7184](https://arxiv.org/abs/1211.7184) [cs.NE]
28. Oliveto, P.S., Witt, C.: Improved time complexity analysis of the simple genetic algorithm. *Theoret. Comput. Sci.* **605**, 21–41 (2015)
29. Prügel-Bennett, A.: Benefits of a population: five mechanisms that advantage population-based algorithms. *IEEE Trans. Evol. Comp.* **14**, 500–517 (2010)
30. Richter, J.N., Wright, A., Paxton, J.: Ignoble trails - where crossover is provably harmful. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) *PPSN 2008*. LNCS, vol. 5199, pp. 92–101. Springer, Heidelberg (2008)
31. Storch, T., Wegener, I.: Real royal road functions for constant population size. *Theor. Comput. Sci.* **320**, 123–134 (2004)
32. Sudholt, D.: Crossover is provably essential for the Ising model on trees. In: *Proceedings of GECCO 2005*, pp. 1161–1167 (2005)
33. Sudholt, D.: Crossover speeds up building-block assembly. In: *Proceedings of GECCO 2012*, pp. 689–702 (2012)
34. Sudholt, D., Thyssen, C.: A simple ant colony optimizer for stochastic shortest path problems. *Algorithmica* **64**, 643–672 (2012)
35. Watson, R.A., Jansen, T.: A building-block royal road where crossover is provably essential. In: *Proceedings of GECCO 2007*, pp. 1452–1459 (2007)
36. Weisstein, E.W.: Erfc, From MathWorld-A Wolfram Web Resource (2015). <http://mathworld.wolfram.com/Erfc.html>
37. Witt, C.: Optimizing linear functions with randomized search heuristics - the robustness of mutation. In: *Proceedings of STACS 2012*, pp. 420–431 (2012)