Ignoble Trails - where crossover is provably harmful *

J. Neal Richter[1], Alden Wright[2], and John Paxton[1]

[1] Montana State University,
richter@cs.montana.edu, paxton@cs.montana.edu,
[2] University of Montana,
alden.wright@umontana.edu,

Abstract. Beginning with the early days of the genetic algorithm and the schema theorem it has often been argued that the crossover operator is the more important genetic operator. The early Royal Road functions were put forth as an example where crossover would excel, yet mutation based EAs were subsequently shown to experimentally outperform GAs with crossover on these functions. Recently several new Royal Roads have been introduced and proved to require expected polynomial time for GAs with crossover, while needing exponential time to optimize for mutation-only EAs. This paper does the converse, showing proofs that GAs with crossover require exponential optimization time on new Ignoble Trail functions while mutation based EAs optimize them efficiently.

1 Introduction

First proposed by Mitchell et al. [1], the well known Royal Road class of fitness functions were designed to demonstrate the essential nature of the crossover operator in genetic algorithms in optimizing that class of fitness functions. They also showed that for an idealized GA ignoring the effects of hitchhiking, the expected optimization time is $O(2^k \log(n/k))$. Somewhat unexpectedly, follow up experimental studies by Forrest and Mitchell [2] show that some random mutation hill-climbers outperform GAs with crossover on the Royal Road. This prompted the same authors to define an open problem in [3].

– Define a family of functions and prove that genetic algorithms are essentially better than evolutionary algorithms without crossover.

In [4] Jansen and Wegener proved that the expected optimization time of the well known (1+1) EA on the classic Royal Road function is $O(2^k(n/k)\log(n/k))$ where n is the string length, k is the length of sub elements of the string and $1/n$ is the mutation rate. Recently in EA research there have been several fitness functions built to meet this challenge in a rigorous way and are discussed in the next section. The goal of this paper is to do the opposite, to provide a fitness

* FINAL REPRINT DRAFT ver 1.99 - Mon Jun 23 22:00:00 HST 2008

function where EAs with mutation alone are provably better at optimization than GAs with crossover. We are not alone in seeking this result. Very recently Poli et al. have produced a fitness function called OneMix [5] where crossover is shown experimentally to be not helpful.

2 Functions Hard for Mutation based EAs

In this section we highlight past research on fitness functions hard to optimize with mutation alone, while being much easier with the use of crossover.

The concatenated trap functions of Deb and Goldberg [6] consist of many concatenations of smaller fully deceptive fitness functions. In a fully deceptive function, all points in the space other than the optima give local advice to go in the opposite direction of the optima. By concatenating these functions together, they set up a situation to illustrate where the building block hypothesis [6] shines. Mutation fails to optimize the function, while the crossover operator builds short order sequences of highly fit bits and recombines these sequences to successfully optimize the function.

One explanation for the failure of Royal Road functions to demonstrate the necessity for crossover is that Royal Road functions are both separable and non-deceptive. Watson [7, 8] created a hierarchical fitness function called HIFF where sub-blocks are interdependent and non-separable as well as being deceptive to the mutation operator. Non crossover EAs require expected exponential time to optimize the HIFF. Dietzfelbinger et al. [9] asymptotically analyzed a recombinative hill-climber on the HIFF function and showed an expected time complexity of $\Theta(n \log n)$.

Jansen and Wegener [10] showed a unitation fitness function called $\text{JUMP}_{m,n}$ with a deceptive quality. The function contains a false optimum with a neighboring fitness canyon (of size $m < n$) with the true optimum on the other side of the canyon. A steady-state hill-climber that accepts no fitness decreases, like the (μ+1) EA, must simultaneously mutate m bits to cross the canyon. The waiting time for this event is $O(n^m)$, while the waiting time for a steady state GA (with uniform crossover) to optimize $\text{JUMP}_{m,n}$ is $O(n^2 \log n)$ steps.

Jansen and Wegener [4] followed up by introducing Real Royal Road functions where a steady state GA with both uniform and one-point crossover have polynomial expected time. EAs without crossover take expected exponential time to optimize these functions.

Storch and Wegener [11] showed additional Real Royal Roads for both uniform and one-point crossover. Using the (2+1) GA with crossover they proved that these fitness function are optimized in expected polynomial time, while the (2+1) EA will take expected exponential time. The (2+1) EA and GA are redefined in later sections.

One critique of the Real Royal Roads is that they are artificial constructs designed to prove a point. Responding, other researchers have produced works on more natural functions. Fischer and Wegener [12] show a mixed result in the one-dimensional Ising Model where for a correctly chosen λ, the $(1 + \lambda)EA$

performs well compared to typical GAs. They do prove that a specialized GAs do far better than the EA for both one and two point crossover. Sudholt [13] shows that a GA requires polynomial time to optimize another Ising Model, while the EA requires expected exponential time.

Finally, Doerr et al. [14] have an upcoming paper showing that crossover has a provable modest advantage one a real world all pairs shortest path graph problem.

3 Minimal Population Evolutionary Algorithms

These algorithms are instances of steady-state evolutionary algorithms [15] where the population is not fully replaced at each generation. A no-duplicates policy is also in place, forcing a population of distinct strings.

3.1 The steady-state (2+1) EA

Here we restate the (2+1) EA. It is an instance of the well-known (μ+1) EA, studied among other places in [16].

Algorithm 1: The (2+1) EA

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

3.2 The steady-state (2+1) GA

Here we redefine the simple steady-state GA from [11] that works on a population size of 2, the smallest population size allowing crossover. Note that the usage of equal probability $\frac{1}{3}$ in the search step is arbitrary. The later results hold for any constant probability ϵ where $\epsilon > 0$.

Algorithm 2: The (2+1) GA

1. *Initialization*: Randomly choose two different individuals $x, y \epsilon \{0,1\}^n$.
2. *Search*: Produce an individual z,
   - with probability 1/3, z is created by mutate(x),
   - with probability 1/3, z is created by mutate(y),
   - with probability 1/3, z is created by mutate(crossover(x, y)).

3. *Selection*: Create the new population $\mathscr{P}$.
   - If $z = x$ or $z = y$, then $\mathscr{P} := \{x, y\}$
   - Otherwise, let $a \epsilon \{x, y, z\}$ be randomly chosen among individuals with the worst f-value. Then $\mathscr{P} := \{x, y, z :\} - \{a\}$.
4. Goto Search

4 Ignoble Trails

We now define a new class of functions called Ignoble Trails. These functions are created for the purpose of rigorously proving that a given mutation based EA outperforms a given crossover based GA on these functions. Like the Real Royal Roads and the HIFF functions, they are somewhat contrived to serve a specific theoretical purpose. We make no claim that real world problems can be mapped to these new functions.

4.1 Ignoble Trails vs Uniform Crossover

The first function $IT1_n^u(x)$ is a modification of the $R_n^u(x)$ function of [11] for uniform crossover. The symbol u refers to the uniform crossover operator. Most of the details are the same as $R_n^u(x)$ except for the addition of b^{**}. Assume a bit-string length of $n := 6m$, where n and m are even integers. Also note that $\|x\|$ refers to the number of ones in the string, $|x|$ is the length in bits of x, and $H(x, y)$ is the Hamming distance of x and y.

$$IT1_n^u(x) := \begin{cases} 16m & x = b^{**} \\ 15m & x \in T \\ 14m & x = a^* \\ 6m + i & x = a_i \in P_1 \cup P_2 \\ 6m - \|x\| & x \in R := \{0,1\}^n - P - T - \{b^{**}\} \end{cases}$$

The major features of $IT1_n^u(x)$ are as follows. The base fitness of the set R is defined to slope in increasing fitness towards the all zeros string. The path P is a sequence of distinct strings $a_1, ..., a_p$ such that consecutive strings on the path have a Hamming distance of 1. P contains $7m+1$ total points where $a_i = 0^{n-i}1^i$ for $i \leq 6m$, and $a_i = 1^{n-j}0^j$ for $i = 6m + j$. P is segmented into two subpaths P_1 and P_2.

The P_1 subpath is defined as points $(a_0, ..., a_{5m-1})$ and the P_2 subpath is defined as (a_{5m+1}, a_{7m}). The fitness for the total path is $6m + i$, with the single exception that a local optimum is created at point $a^* := a_{5m}$ with fitness $14m$. The other local optimum of P is at the endpoint $a^{**} := a_{7m}$ with fitness value $13m$.

There also exists an area T defined to contain all points $b1^{4m}c$ where the substrings b and c obey $|b| = |c| = m$ and $||b|| = ||c|| = m/2$. In $R_n^u(x)$ T is the target and can be created with high probability with a population of $\{a^*, a^{**}\} := \left\{0^m 1^{5m}, 1^{5m} 0^m\right\}$ via uniform crossover.

Our crucial modification to $R_n^u(x)$ is to add a point b^{**} with fitness greater than the region T. This point is defined as a point with k bits different than a^{**}, or $H(a^{**}, b^{**}) = k$. Here k is defined to be a constant where $n = 6m$ is chosen so that $3 < k < m/4$. We define b^{**} to be $1^m 0^k 1^{4m-k} 0^m$.

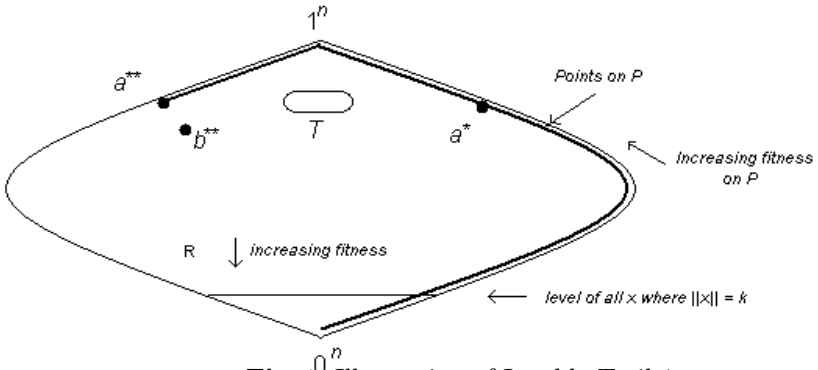

Fig. 1. Illustration of Ignoble Trail 1

4.2 Behavior of the EA and GA on Ignoble Trail 1

Referring to Figure 1, the initial random population of two distinct individuals will begin the process of traveling down R towards the initial point of P, $P_0 := 0^n$. Both algorithms will discover and optimize P unless exceptional luck strikes and $\{T \cup \{b^{**}\}\}$ is discovered first. Since the selection method prohibits duplicate strings, once they are on path P there is a leading point and a trailing point on P. They travel up P until such time as a^* is found [there is a probability $\Theta(1/n)$ a^* is skipped]. If a^* is found, the behavior degenerates to mimic the $(1+1)$ EA as a^* is fixed in the population and the other string is available for continued optimization of P until a^{**} is found.

Once the population becomes $\{a^*, a^{**}\}$ the behavior of the two algorithms diverges. The EA is very unlikely to discover T via mutation, and is likely to find b^{**} in $O(n^k)$ steps. Conversely the GA is very likely to discover T via crossover before it discovers b^{**}. Once the GA has found T, it will accumulate both individuals in T in short order. The expected waiting time to discover b^{**} from T is exponential. Thus we refer to T as the 'trap' rather than the 'target' of $R_n^u(x)$. Note that crossover is of little assistance in discovering b^{**} from either a^{**} or T.

Figure 2 contains a visual representation of the results to follow and the high likelihood optimization phases of both algorithms.

4.3 Time Complexity Results

Note that the next set of proofs take some arguments from [11] or [4]. The addition of b^{**} requires many additional steps to prove rigorous results, there are many more good and bad events to account for above those from [11].

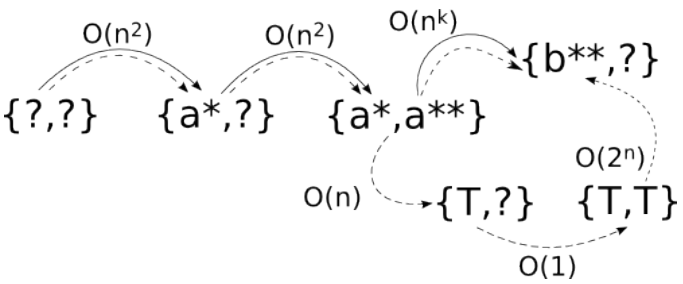

Fig. 2. Diagram of proofs of Lemmas and Theorems for $IT1_n^u(x)$ - Solid lines are events associated with the $(2+1)$ EA, dashed lines are events associated with the $(2+1)$ GA. The labels on each arc refer to the expected waiting time to transition from state to state.

Lemma 1. *The probability that* $(2+1)$ *EA without crossover and the* $(2+1)$ *GA with uniform crossover find a point in* $P_2 \cup T \cup \{b^{**}\}$ *without discovering path* P_1 *within* $O(n^2)$ *steps is at least* $1 - e^{-\Omega(n)}$.

Proof. Recall that k is a constant, and assume that $n = 6m$ is chosen so that $3 < k < m/4$. Let $Q := P_2 \cup T \cup \{b^{**}\}$ and note that all elements of Q have at least $5m - k$ ones. Let R be the set of points not in P with at most $4m$ ones. The probability of initializing a member of the population with more than $4m$ ones is $e^{-\Omega(n)}$ by Chernoff's bound [17]. Since Q is contained in that same set, the same holds for Q. Each point of R has a better Hamming neighbor. The probability of discovering that neighbor via mutation is at least $p = 1/(3en)$. Applying Chernoff bounds, the waiting time for at most $n = 6m$ successful events is $O(n^2)$, and the probability that this waiting time is exceeded is exponentially small. The probability of producing a point in Q from R via mutation is at most $n^{-m+k} = e^{-\Omega(n)}$ by Chernoff's bound. Turning to the crossover operator, the probability of producing a point in Q from two points in R via crossover is $e^{-\Omega(n)}$ by the following argument. Let d be the Hamming distance between the two parent strings r_1 and r_2. Let $s = ||r_1 \wedge r_2||$, thus the expected number of ones is $s + d/2$. Unless $d > m - k$, the child string can not have at least $5m - k$ ones. Applying Chernoff's bound on the differing bits of the parents, $r_1 \oplus r_2$, the probability to create at least $d/2 + m - k$ ones is $e^{-\Omega(n)}$. As for the joint operator, the probability of producing a point in Q from two points in R via crossover and mutation is $e^{-\Omega(n)}$ as follows. Either crossover produces a point with at least $9m/2 - k$ ones or it doesn't. In the first case, the probability that crossover produces a point with at least $9m/2 - k$ ones is $e^{-\Omega(n)}$ by the Chernoff bounds on the the bits differing in the parents. In the other case, mutation must go from a point with less than $9m/2 - k$ ones to a point with at least $5m$ ones, and the probability that this happens $n^{-m/2+k} = e^{-\Omega(n)}$. Applying the union bound, we see that the total failure probability is $e^{-\Omega(n)}$. □

Lemma 2. *The* $(2+1)$ *EA will optimize* P *and find* $\{a^{**} \cup b^{**}\}$ *in* $O(n^2)$ *steps with probability* $1 - 2^{-\Omega(n)}$. *The* $(2+1)$ *EA will discover a point in* T *from* P *with probability* $2^{-\Omega(n)}$.

Proof. Beginning from Lemma 1, we assume the population contains a point in P_1. Each point on the path P has a better fitness Hamming neighbor. The probability of discovering that neighbor via mutation is at least $p = 1/(3en)$. Inverting and substituting we get a waiting time of at most $7m$ (the length of P) successful events of probability p. Applying Chernoff's bound we get the first result above. As for the second result, by the definitions of P and T, the Hamming distance between them is at least $m/2$. The mutation hitting probability is $(1/n)^{m/2}(1-1/n)^{n-m/2}$. However, there are $\binom{m}{m/2}^2$ points in T, so the probability of hitting T is increased by this amount. Bounding the number of points in T via a standard binomial coefficient inequality [3], we get $\binom{m}{m/2}^2 \leq (2e)^{m/2}$. Thus we bound the probability of hitting T from $p_i \in P$ by $(1/n)^{m/2}(1-1/n)^{n-m/2}(2e)^{m/2} \leq (2e/n)^{m/2} < 2^{-\Omega(n)}$. □

Theorem 3. *The (2+1) EA without crossover will optimize the* $IT1_n^u(x)$ *function in expected* $O(n^k)$ *steps and within* $O(n^k \ln k)$ *steps with probability* $1 - O(1/n)$.

Proof. Referring to Lemma 2, the next step is to establish the expected waiting time to discover b^{**} from a population of $\{a^{**}, p_i \in P\}$. The Hamming distance between a^{**} and b^{**} is defined to be constant k where $n = 6m$ is chosen so that $3 < k < m/4$. Thus the probability of mutating from a^{**} to b^{**} in one step is $p = (1/n)^k(1-1/n)^{n-k}$. This is bounded below by $1/(en^k)$, resulting in an expected waiting time that is bounded above by $en^k = \Theta(n^k)$. Note that this is the best case possibility of finding b^{**} from any point on P as the Hamming distance for all points in P is $H(p_i \in P, b^{**}) \geq k$. Applying Chernoff bounds, the probability of finding b^{**} within $en^k \ln k$ steps is $1 - O(1/n)$. From Lemma 2 we know that the probability of finding T from any point in P is exponentially small. Thus the probability of finding T before finding b^{**} is also exponentially small. □

Lemma 4. *The* $(2+1)$ *GA with uniform crossover will discover a point in* $P_2 \cup T \cup \{a^*\}$ *in* $O(n^2)$ *steps with probability* $1 - 2^{-\Omega(n)}$. *The probability of the* $(2+1)$ *GA with uniform crossover finding* $\{b^{**}\}$ *while searching for* $P_2 \cup T \cup \{a^*\}$ *is* $2^{-\Omega(n)}$.

Proof. Lemma 2 of [11] proves the first part of the result. For the second result, note that b^{**} contains $5m - k$ ones. Recall that k is a constant, and assume that $n = 6m$ is chosen so that $3 < k < m/4$. We have already shown that as long as the points in the population contain at least $4m$ ones, the probability of finding b^{**} is exponentially small. The remaining possibility is mutating to b^{**} from a point in the population $p_i \in \{P_1 - a^*\}$ where $a_{4m} < p_i < a_{5m}$. It is easy to see

[3] $\binom{n}{k} \leq \left(\frac{en}{k}\right)^k$

that it is exponentially unlikely that the other point of the population is not in $\{a_i \in P_1 \; i \geq m\}$. The minimum Hamming distance between a point of the population and b^{**} is $2m-k$, so the probability of finding b^{**} by mutation is at most $2^{-\Omega(n)}$. Turning to the crossover operator, recall that $b^{**} = 1^m 0^k 1^{4m-k} 0^m$. Both members of the population are of the form $0^{n-i}1^i$ for $m \leq i < 5m$ so both points have 1s in the last m positions. Thus, it is impossible to cross the two points in the population to produce a point with Hamming distance less than m from b^{**}. □

Proposition 5. *With probability $O(1/n)$, the $(2+1)$ GA will find a point in $P_2 \cup T \cup \{b^{**}\}$ before finding a^*.*

Proof. The proof of Theorem 4 of [11] shows this result without reference to b^{**}. The Hamming distance from P_1 to b^{**} is exponential, and thus does not change the result. □

Lemma 6. *If the population contains a^*, the $(2+1)$ GA will find a point in T in $O(n^2)$ steps with probability $1 - O(1/n^k)$.*

Proof. Lemma 3 of [11] shows the result with probability $1 - 2^{-\Omega(n)}$. We must consider the possibility that b^{**} is found before T. To start, we consider the possibility of finding b^{**} by crossover plus mutation from a population of a^* and any other point $a_i \in P$. For $0 \leq i \leq 5m$ this is exponentially unlikely via the argument given in the proof of Lemma 4. For $5m < i \leq 6m$ this results in crossover on a^* and a_i setting the last m positions to 1. Yet b^{**} has zeros in these positions, so subsequent mutation must flip at least m bits. Finally, if the other point is $a_{6m+j} = 1^{n-j}0^j$ for $0 < j \leq m$, then a^* and a_{6m+j} agree in $k+m-j$ bits different from the corresponding bits of b^{**}. Thus crossover and subsequent mutation of at least those $k+m-j$ bits is required, giving a probability of discovering b^{**} bounded above by $O(1/n^{k+m-j})$. As long as a_i is not a^{**}, a better point on P will be discovered with probability $1/(3en)$. From this and the bounds derived above, we can see that either a^{**} or a point of T will be found with probability $1 - O(1/n^k)$.

Now assume the population $\{a^*, a^{**}\}$. The one-step probability of finding b^{**} by either mutation or crossover followed by mutation is $p = O(1/n^k)$ whereas the one-step probability of discovering T was shown to be $q = \Theta(1/n)$ in Lemma 3 of [11] by an application of Sterling's formula. There is a sequence of independent trials until one or the other of these outcomes happens. A probability argument[4] shows that the probability of finding b^{**} over all trials is $p/(p+q) = O(1/n^k)/(O(1/n^k)+O(1/n)) = O(1/n^k)/O(1/n) = O(1/n^{k-1})$. □

Lemma 7. *The expected waiting time to hit b^{**} from a population $\{t_i \in T, t_j \in T\}$ is exponential for the (2+1) GA with uniform crossover.*

[4] Given that either event A or B will eventually happen, let $p := Pr[A]$, $q := Pr[B]$ and $r := 1-p-q$. The probability that A eventually happens is $p/(1-r) = p/(p+q)$.

Proof. It is possible for a crossover plus mutation operation to get b^{**} from two elements of T. Remember that $b^{**} := 1^m 0^k 1^{4m-k} 0^m$. If the two population elements of T are binary complements of each other in the b and c regions, and if the crossover mask is chosen correctly, crossover could get the first and last m bits of the child to match b^{**}. Then mutation would need to get the k bits of the middle 1^{4m} bits to match b^{**}. The probability of getting the correct crossover mask is 2^{-2m}. Thus the probability of getting the correct mask and the correct mutation is bounded above by $O(2^{-2m})$.

Another possibility would be for crossover to get all but $0 \leq j \leq 2m$ of the first and last m bits correct. These correspond to the substrings b and c from the definition of T, $b1^{4m}c$ where b and c contain exactly half 1s. It is not necessary for these j bits of the crossover mask to be correct, thus the probability of choosing the correct crossover mask is 2^{-2m+j}. Following crossover, mutation must correct $k+j$ bits, with probability $(1/n)^{k+j}(1-1/n)^{n-k-j} \leq (1/n)^{k+j}$. Consequently, the probability of getting the crossover mask right and the correct mutation is $\leq (1/n)^{k+j}(1/2)^{2m-j} \leq (1/2)^{2m+j}$ which is exponentially small. □

Theorem 8. *The (2+1) GA with uniform crossover will need exponential time steps to optimize $IT1_n^u(x)$ with probability $1 - O(1/n)$.*

Proof. Beginning from Prop. 5 and Lemma 6 above, assume the population contains a point in T. By the selection method of the GA, once a member of T exists in the population we should only have to wait constant time $O(1)$ for both members of the population to be in T. Once the GA contains two members of T, probability of crossover plus mutation or mutation alone discovering b^{**} is exponentially unlikely by Lemma 7. Of the various bad events, the probability from Prop. 5 of skipping a^* is maximal at $O(1/n)$. □

5 Conclusions

We believe we have shown for the first time a proven example of a situation where a crossover based GA is expected to be exponentially outperformed by an EA without the crossover operator. Future work will expand upon this result with empirical studies and extensions to cover 1-point crossover. In addition it is believed that examples can be created for large population EA/GAs showing this exponential performance difference. An open problem would be to follow up on [14] and produce a reasonable graph problem that where the GA is outperformed by an EA.

6 Acknowledgments

The authors would like to thank Thomas Jansen, Ingo Wegener, Tomas Gedeon and the anonymous reviewers for their encouragement and advice. We would also like to thank the participants and organizers of the Theory of Evolutionary Algorithms conference at Schloss Dagstuhl International Conference and Research Center for Computer Science, where the seeds of this research were germinated.

References

1. M. Mitchell, S. Forrest, and J. H. Holland, "The royal road for genetic algorithms: Fitness landscapes and GA performance," in *Proceedings of the First European Conference on Artificial Life. Toward a Practice of Autonomous Systems* (F. J. Varela and P. Bourgine, eds.), (Cambridge, MA), pp. 243–254, MIT Press, 1992.
2. S. Forrest and M. Mitchell, "Relative building-block fitness and the building-block hypothesis," in *Foundations of genetic algorithms 2* (L. D. Whitley, ed.), (San Mateo), pp. 109–126, Morgan Kaufmann, 1993.
3. M. Mitchell, J. H. Holland, and S. Forrest, "When will a genetic algorithm outperform hill climbing.," in *NIPS* (J. D. Cowan, G. Tesauro, and J. Alspector, eds.), pp. 51–58, Morgan Kaufmann, 1993.
4. T. Jansen and I. Wegener, "Real royal road functions–where crossover provably is essential," *Discrete Applied Mathematics*, vol. 149, no. 1-3, pp. 111–125, 2005.
5. R. Poli, A. H. Wright, N. McPhee, and W. Langdon, "Emergent behaviour, population-based search and low-pass filtering," in *Proceedings of the Congress on Evolutionary Computation (CEC) 2006*, pp. 88–95, IEEE Press, 2006.
6. K. Deb and D. E. Goldberg, "Analyzing deception in trap functions," in *FOGA* (L. D. Whitley, ed.), pp. 93–108, Morgan Kaufmann, 1992.
7. R. A. Watson, "Analysis of recombinative algorithms on a non-separable building-block problem," in *Foundations of Genetic Algorithms 6*, pp. 69–89, Morgan Kaufmann, 2001.
8. R. A. Watson and J. B. Pollack, "Hierarchically consistent test problems for genetic algorithms," in *Proceedings of the Congress on Evolutionary Computation* (P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, eds.), vol. 2, (Mayflower Hotel, Washington D.C., USA), pp. 1406–1413, IEEE Press, 6-9 1999.
9. M. Dietzfelbinger, B. Naudts, C. V. Hoyweghen, and I. Wegener, "The analysis of a recombinative hill-climber on h-iff," *IEEE Trans. Evolutionary Computation*, vol. 7, no. 5, pp. 417–423, 2003.
10. T. Jansen and I. Wegener, "The analysis of evolutionary algorithms - a proof that crossover really can help," *Algorithmica*, vol. 34, no. 1, pp. 47–66, 2002.
11. T. Storch and I. Wegener, "Real royal road functions for constant population size," *Theor. Comput. Sci.*, vol. 320, no. 1, pp. 123–134, 2004.
12. S. Fischer and I. Wegener, "The one-dimensional ising model: Mutation versus recombination," *Theor. Comput. Sci.*, vol. 344, no. 2-3, pp. 208–225, 2005.
13. D. Sudholt, "Crossover is provably essential for the ising model on trees," in *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, (New York, NY, USA), pp. 1161–1167, ACM, 2005.
14. B. Doerr, E. Happ, and C. Klein, "Crossover is provably useful in evolutionary computation," in *GECCO '08: Proceedings of the 2008 conference on Genetic and evolutionary computation*, vol. to appear, 2008.
15. L. Davis, *Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhold, 1991.
16. G. Rudolph, *Convergence Properties of Evolutionary Algorithms*. Hamburg: Dr. Kovac, 1997.
17. R. Motwani and P. Raghavan, *Randomized algorithms*. New York, NY, USA: Cambridge University Press, 1995.