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

The Analysis of a Recombinative Hill-Climber on HIFF

Martin Dietzfelbinger, Bart Naudts,
Clarissa van Hoyweghen, and Ingo Wegener

No. CI-138/02

Technical Report ISSN 1433-3325 August 2002

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.

The Analysis of a Recombinative Hill-Climber on H-IFF

Martin Dietzfelbinger[1], Bart Naudts[2], Clarissa van Hoyweghen[2], and Ingo Wegener[3*]

Abstract— **Many experiments have proved that crossover is an essential search operator in evolutionary algorithms, at least for certain functions. However, the rigorous analysis of such algorithms on crossover-friendly functions is still in its infancy. Here a recombinative hill-climber is analyzed on the crossover-friendly function H-IFF introduced by [10]. The dynamics of this algorithm are investigated and it is proved that the expected optimization time equals $\Theta(n \log n)$.**

Keywords— **Evolutionary algorithms, recombinative hill-climbers, expected optimization time, one-point crossover.**

I. INTRODUCTION

The class of evolutionary algorithms contains evolution strategies (mainly based on the mutation operator) and genetic algorithms (mainly based on the crossover operator). The experimental knowledge about the behavior of evolutionary algorithms is immense. For many problems these experiments have shown that crossover is essential for an efficient optimization. However, rigorous analytical results not based on any assumptions are rare.

We analyze evolutionary algorithms without stopping criterion as infinite stochastic processes. Then we are interested in the following parameters:

– given a point of time t and a schema s, what is the probability that an individual of the generation at time t contains s,

– for X, the random variable describing the optimization time or first hitting time defined by the minimal t such that the generation at time t contains an optimal individual, we are interested in estimates of $E(X)$, the expected value of X called expected optimization time.

The real royal road functions introduced and analyzed by Jansen and Wegener [3] are up to now the only functions where it has been proved that the expected optimization time of mutation-based evolutionary algorithms is exponential while a generic steady-state genetic algorithm without special mechanisms to ensure diversity within the population has a polynomially bounded expected optimization time. These functions have not a clear building-block structure and the success of the GA cannot be explained by schema theory.

[1]Fakultät für Informatik und Automatisierung, TU Ilmenau, D-98694 Ilmenau, Germany, `dietzfelbinger@theoinf.tu-ilmenau.de`

[2]Department of Mathematics and Computer Science, University of Antwerpen (RUCA), B-2020 Antwerpen, Belgium, `bart.naudts`, `hoyweghe@ruca.ua.ac.be`

[3]FB Informatik, LS 2, Univ. Dortmund, D-44221 Dortmund, Germany, `wegener@ls2.cs.uni-dortmund.de`

*The work of this author was supported by the Deutsche Forschungsgesellschaft (DFG) as part of the Collaborative Research Center “Computational Intelligence” (SFB 531).

Here we analyze a specific evolutionary algorithm on the function H-IFF (hierarchical-if-and-only-if) introduced by Watson *et al.* [10] as an example of a hierarchically decomposable problem with a clear block structure. In Section II we define this function and discuss some of its properties and in Section III we present the recombinative hill-climber we analyze later. It turns out that the optimization of H-IFF by the recombinative hill-climber can be modeled by a simple combinatorial game on a completely balanced binary tree. This makes the analysis easy to describe. Section IV contains some remarks on the dynamics of the optimization process. In Section V, we use a global approach to obtain lower and upper estimates of the expected optimization time which differ asymptotically only by a factor of 2.246. This very precise result is obtained by applications of the coupon collector's theorem (see Motwani and Raghavan [4]) and a delay sequence argument (such arguments were first used by Ranade [6]). We believe that these methods will turn out to be useful also in further investigations of evolutionary algorithms. Finally, we compare in Section VI our asymptotic results with experimental results for problem dimensions between 64 and 4096. We finish with concluding remarks.

II. THE FUNCTION H-IFF AND SOME OF ITS PROPERTIES

The family of functions called H-IFF contains pseudo-boolean functions $\text{H-IFF}_n : \{0,1\}^n \to \mathbb{R}^+$ for $n = 2^k$, k a positive integer. The most illustrative definition of H-IFF_n uses a completely balanced binary tree with n leaves “colored” from left to right with the bits of the input vector $a = (a_1, \ldots, a_n)$, see Figure 1. Then we consider the following coloring rules on the tree.

An inner node gets colored if both children are colored by the same color. Then it gets the same color as its children. Hence, a node v gets colored if and only if all leaves of the subtree rooted at v have the same color. Now we define the value of $\text{H-IFF}_n(a)$. A colored node v has a value which equals the number of leaves of the subtree rooted at v. We number the levels of the tree from the leaves (level 0) to the root (level k). Then the value of a colored node at level l equals 2^l. In our example, $\text{H-IFF}_{16}(a) = 16+14+8+0+0 = 38$, since the leaves on level 0 contribute 16 to the H-IFF value, 7 nodes on level 1 contribute 2 each, 2 nodes on level 2 contribute 4 each, and the nodes on level 3 and level 4 (the root) contribute nothing, since they are not colored.

Each inner node of the tree represents a building block of H-IFF. Subtrees where all leaves are colored by the same color or bit are building blocks and this is indicated by a

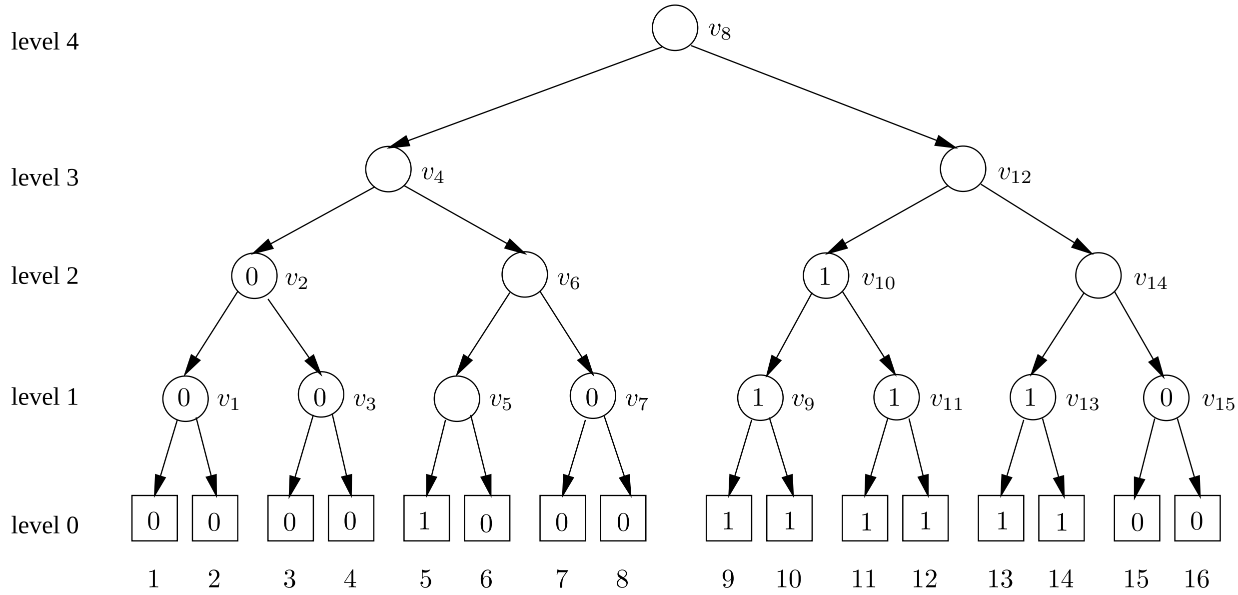

Fig. 1. Illustration of the function H-IFF$_{16}$.

coloring of the root of the subtree. Obviously, the function H-IFF$_n$ has two optima, the all-ones string 1^n and the all-zeros string 0^n. H-IFF functions have the property of *spin-flip* or *bit-flip symmetry* (see [5] in the context of GAs) defined by

$$\text{H-IFF}_n(a) = \text{H-IFF}_n(\bar{a})$$

for the bitwise complement $\bar{a}$ of a. This implies that also the average fitness of a schema $s \in \{0, 1, *\}^n$ is equal to the average fitness of the schema's complement $\bar{s}$ where $\bar{s}_i = 0$, if $s_i = 1$, $\bar{s}_i = 1$, if $s_i = 0$, and $\bar{s}_i = *$, if $s_i = *$.

We discuss how typical randomized search heuristics work on H-IFF functions and a random initial string. As long as a node v is not colored, the fitness is decomposed between the two subtrees rooted at the children of v. Hence, the children of v obtain their colors independently, each color with probability $1/2$. This implies that there will be many nodes with large subtrees where the two children of the root get different colors. Mutations can disturb the colorability of subtrees and there is only a tiny chance that all leaves of one subtree get the opposite color while all leaves of the other subtree keep their color. Hence, it is not too difficult to prove that mutation-based evolutionary algorithms have an exponential expected optimization time on H-IFF functions.

However, there seems to be a good chance for 1-point crossover, e.g., from $0^{n/2}1^{n/2}$ and $1^{n/2}0^{n/2}$ we can obtain 0^n and 1^n. In order to gain from 1-point crossover we need diversity in the population. Whenever all individuals in the population have ones in one block and zeros in the neighboring block, crossover cannot help to escape from this trap. There are recombination-based GAs which apply techniques to maintain diversity in their population, e.g., by using a niching technique (see [7], [8], [10]). This works well for certain functions. Here we investigate a special algorithm.

III. A RECOMBINATIVE HILL-CLIMBER

As we have seen it is essential to save the different alleles at each position. This is guaranteed if we include the bitwise complement $\bar{a}$ of each individual a in the population. Usually one allows mutation which flips each bit independently with probability $1/n$. Then no bit is flipped with a probability of $(1 - 1/n)^n \approx e^{-1}$. Since, with large probability, mutation is useless for H-IFF functions, we consider only one-point crossover. One-point crossover with cut position $m \in \{1, \ldots, n-1\}$ produces $(a_1, \ldots, a_m, b_{m+1}, \ldots, b_n)$ and $(b_1, \ldots, b_m, a_{m+1}, \ldots, a_n)$ from $a = (a_1, \ldots, a_n)$ and $b = (b_1, \ldots, b_n)$. The cut position is chosen randomly. Since we believe that H-IFF functions are easy when using one-point crossover and guaranteeing diversity, we choose the smallest possible population size, namely 2, and use a randomized hill-climber. This leads to the following algorithm.

Algorithm 1

1. Initialization. Choose $x \in \{0, 1\}^n$ randomly according to the uniform distribution and let the initial generation consist of x and its bitwise complement $\bar{x}$.
2. Loop. Choose a random cut position $m \in \{1, \ldots, n-1\}$ according to the uniform distribution and create y by 1-point crossover from x and $\bar{x}$ using cut position m. Replace $(x, \bar{x})$ with $(y, \bar{y})$ iff $\text{H-IFF}_n(y) > \text{H-IFF}_n(x)$.

This recombinative hill-climber is a reduced version of the *Gene Invariant* GA (GIGA) introduced by Culberson [1]. A crossover between x and $\bar{x}$ is also known as *complementary crossover* [2]. We remark that it makes no difference if we replace the condition $\text{H-IFF}_n(y) > \text{H-IFF}_n(x)$ by the more usual condition $\text{H-IFF}_n(y) \geq \text{H-IFF}_n(x)$. We can consider this algorithm also as (1+1)GA, since the population can be described by the single individual x and we always choose the better one from the parent x (and $\bar{x}$) and its child y (and $\bar{y}$). The difference to the well-known (1+1)EA is that mutation is replaced with one-point crossover. In the following we use the notion of a (1+1)GA.

The tree illustration of the function H-IFF$_n$ (see Figure 1) is also well suited as illustration of the effect of one-point crossover. The cut position m corresponds one-to-one to the inner node v_m of the tree which has the property that all leaves of its left subtree belong to bit positions $i \leq m$ and all leaves of its right subtree belong to bit positions $i > m$. The individual y is obtained from x by negating all bits at positions $i > m$. How do we obtain the coloring of the inner nodes of the y-tree (the coloring of leaf i by y_i) from the x-tree? We distinguish three types of nodes:
– the nodes on the path from the root to v_m including v_m,
– the nodes to the left of this path and in the left subtree of v_m, and
– the nodes to the right of this path and in the right subtree of v_m.
Nodes of the second group do not change their color status or their color, since all leaves below these nodes have the same color as before. Nodes of the third group do not change their color status but their color (if they are colored), since all leaves below these nodes have obtained the opposite color than before. Hence, the nodes of the first group decide whether the step is accepted and we replace x with y. If v_m is colored in the x-tree, it is not colored in the y-tree, since the two subtrees get leaves with different colors. Then the step is not accepted. If v_m is not colored in the x-tree and one of its children is also not colored in the x-tree, v_m remains uncolored, since one child remains uncolored. Again the step is not accepted. If v_m is not colored in the x-tree and both children are colored in the x-tree, then the children are colored in the x-tree by different colors. The right child gets the opposite color in the y-tree implying that both children have the same color in the y-tree and v_m gets colored. Then the step is accepted and x is replaced with y. Note that more nodes on the path from the root to v_m may get colored. Such nodes are called *free-riders,* since such a node v_j gets colored, although we have not chosen the cut position j. In Figure 1, only the uncolored nodes v_5 and v_{14} have two colored children. If we choose position 5, then v_5 gets colored by 1 leading to the free-rider v_6 which gets the color 1, since also v_7 has been recolored by color 1. The node v_4 is not a free-rider, since v_2 is still colored by 0 and v_6 has been colored by 1. If we choose position 14, we color v_{14} by color 1 and v_{12} is a free-rider, since v_{10} is still colored by 1. We cannot color v_8, since the sibling v_4 of v_{12} is not colored.

Hence, the (1+1)GA is equivalent to the following game on the tree.

Game 1 (with free-riders)
1. Start with the x-tree for a randomly chosen $x \in \{0,1\}^n$.
2. Loop. Choose $m \in \{1,\ldots,n-1\}$ randomly. Do nothing if v_m is colored or one of the two children of v_m is not colored. Otherwise, flip the color of all nodes in the right subtree of v_m and all nodes to the right of the path from the root to v_m. Color v_m with the color of its children and for each node v_j on the path from v_m to the root (in this order) test whether v_j's two children are colored with the same color and if so, color also v_j with this color.

The random optimization time of the (1+1)GA on H-IFF$_n$ is equal to the random time when the root of the tree gets colored in Game 1. The (1+1)GA finds a building block at the same time when the corresponding node of the tree gets colored. It is easier to discuss the game than the (1+1)GA. Later, for an upper bound on the expected optimization time we will ignore free-riders. Then we can consider the following game which is equivalent to this situation.

Game 2 (without free-riders)
1. Start with a tree where only the leaves are colored (there is only one color).
2. Loop. Choose $m \in \{1,\ldots,n-1\}$ randomly. Do nothing if v_m is colored or one of the two children of v_m is not colored. Otherwise, color v_m.

Again we are interested in the time until some node and, in particular, the root gets colored.

IV. THE DYNAMICS OF THE (1+1)GA ON H-IFF

We are interested in the probability of having a building block at a certain position, i.e., a block of bit positions $(j-1)2^l+1,\ldots,j\cdot 2^l$, $1 \leq j \leq 2^{k-l}$, of the same value, at time step t. Considering the building block of length n this is the probability of having optimized H-IFF with the (1+1)GA within t time steps. We have seen that our question is equivalent to the question whether a node on level l is colored at time step t by Game 1. It follows directly that this probability is the same for all nodes on level l.

We denote by $P_n(l,t)$ the probability that a node on level m in a tree with $k = \log n$ levels is colored after time step t. It is obvious that $P_n(0,t) = 1$ for all n and t, since the leaves are colored immediately. Also

$$P_n(l,1) = 2 \cdot 2^{-2^l}$$

for all l, since the considered subtree has 2^l leaves which are colored randomly and two of these colorings lead to a coloring of the root of this subtree.

We discuss how to compute $P_n(l,t)$ and fix a node v at level l and investigate the subtree rooted at v.

Node v is colored at time step $t+1$ iff it was colored at time step t (probability $P_n(l,t)$) or it is colored for the first time at time step $t+1$. The latter occurs iff the following holds for some node w on some level $l-i+1$, $1 \leq i \leq l$, situated in the subtree rooted at v (see Figure 2):
– the cut position at time step $t+1$ corresponds to w,
– let $v_0 = v, v_1, \ldots, v_{i-2}, v_{i-1} = w$ be the nodes of the path between v and w, then it holds that the children of $v_0,\ldots,v_{i-1}$ not on this path are colored such that all nodes to the left of the path have the same color and the nodes to the right of the path have the opposite color.

If the path from v_0 and w has these properties, then this path is called a colorable path of length $i-1$ and w is called the source of the path. For each $i \in \{1,\ldots,l\}$ there are 2^{i-1} nodes which are possible sources of colorable paths and 2 possibilities how to choose the color to the left

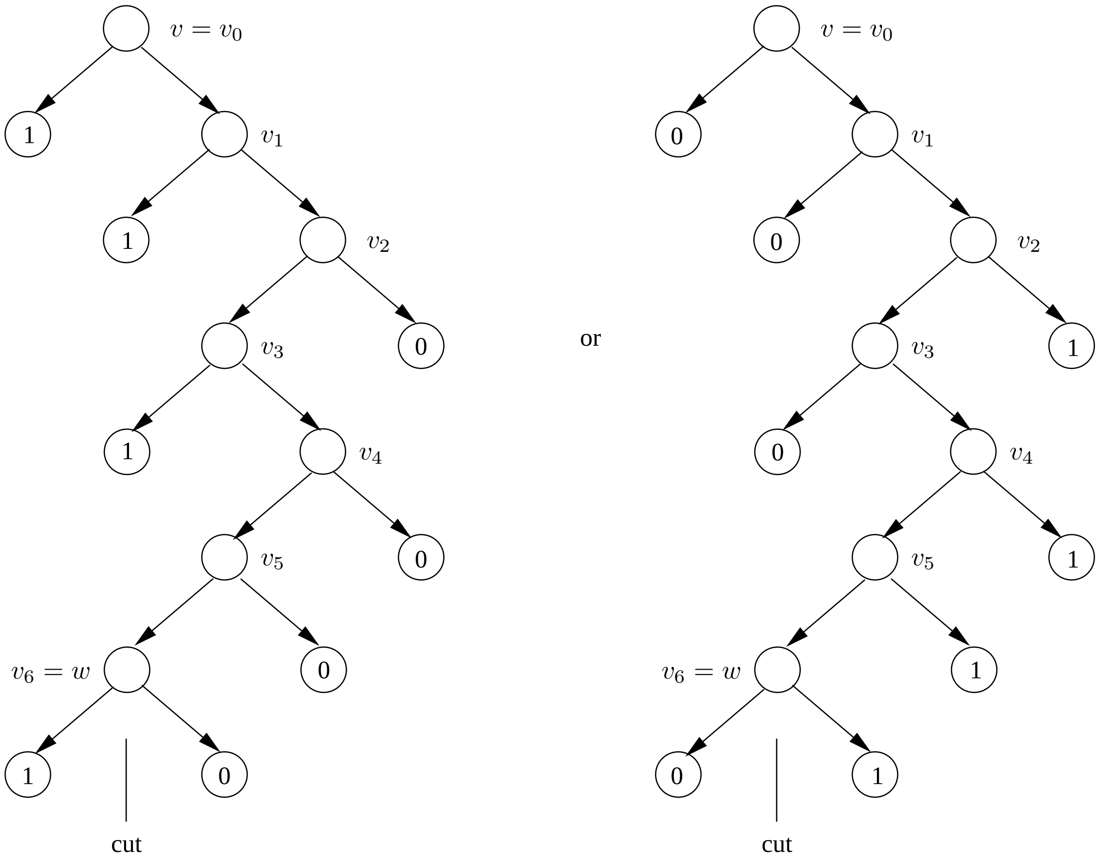

Fig. 2. A situation where v gets colored for the first time.

of the path. Let $Q_n(l,t,i)$ be the probability that a specific path of length $i-1$ starting at a node on level l is at the end of step t a colorable path with a given coloring of the nodes left to the path. Since the probability of choosing the appropriate cut position equals $1/(n-1)$, we obtain

$$P_n(l,t+1) = P_n(l,t) + \frac{1}{n-1} \cdot \sum_{1 \le i \le l} 2^i \cdot Q_n(l,t,i).$$

The event whose probability is defined as $Q_n(l,t,i)$ is the intersection of the following events:

– the left children of the path nodes which are not on the path are colored with the same color c and
– the right children of the path nodes which are not on the path are colored by the opposite color $\bar{c}$.

However, the events that certain nodes are colored are not independent even for the nodes not lying on the same path. Knowing that one node is colored leads to a tendency of many chosen nodes in the corresponding subtree and this implies that is more likely that other nodes have not been colored.

We conclude that it seems to be too difficult to derive a closed formula for $P_n(l,t)$ and we have to be satisfied with an asymptotic analysis.

V. Asymptotically Tight Bounds on the Expected Optimization Time of the (1+1)GA on H-IFF

Stochastic processes such as evolutionary algorithms are in almost all cases too complicated to be analyzed exactly. The theory on deterministic and randomized specialized algorithms has shown that one can get meaningful results only if one looks for estimates of the interesting values. This also holds for the analysis of the (1+1)GA on H-IFF functions. We first prove a lower bound on the expected waiting time until Game 2 is finished. Then it is quite easy to consider the effect of the free-riders to obtain a lower bound on the expected optimization time of the (1+1)GA on H-IFF. We apply the coupon collector's theorem (see [4]) which considers the following scenario. There are N different types of coupons. At each point of time we get a coupon whose type is chosen randomly from $\{1,\ldots,N\}$ and independent from all the coupons chosen before. Let T_N be the time until we have at least one coupon of each type. Then

$$E(T_N) = N\left(1 + \frac{1}{2} + \cdots + \frac{1}{N}\right)$$

$$= N \ln N + O(N)$$

and it is quite unlikely that T_N deviates from $E(T_N)$ by more than cN, for c sufficiently large.

Theorem 1: The expected time of Game 2 is bounded below by $n \ln n + n \log n - O(n) \approx 1.693\, n \log n$.

Proof: We consider two phases where the first phase ends when each node on level 1 is colored. We claim that the expected length of the first phase is bounded below by $n \ln n - O(n)$. If we consider only nodes on level 1 and the steps where such a node is chosen, we are in the scenario of the coupon collector's theorem with $n/2$ coupons and the expected waiting time is bounded below by $(n/2)\ln(n/2) - O(n) = (1/2)n \ln n - O(n)$. However, the probability of choosing such a node equals only $(n/2)/(n-1)$. The expected time until we have chosen in $(1/2)n \ln n - O(n)$ steps such a node is bounded below by $n \ln n - O(n)$.

At the end of the first phase there is a path from the root to a node v on level 1 where only v is colored. The $\log n - 1$ uncolored nodes on this path have to be colored in a fixed order. The expected time until the root is colored is bounded below by $(n-1)(\log n - 1) = n \log n - O(n)$. This finishes the proof. ∎

The lower bound on the expected time until the root is colored also holds with large probability. This is true for the coupon collector's theorem and for the second phase we can apply a Chernoff bound for $n \log n - O(n)$ trials with a success probability of $1/(n-1)$. (For Chernoff bounds see, e.g., [4]).

If we look for a lower bound for the (1+1)GA on H-IFF we have to take free-riders into account. When coloring the leaves we have by a Chernoff bound with overwhelming probability less than $n/4 + O(n^{3/4})$ free-riders on level 1 and, therefore, we have to color at least $n/4 - O(n^{3/4})$ selected nodes on level 1. Again, we can apply a Chernoff bound to obtain a lower bound of $(n/4)\ln(n/4) - O(n) = (1/4)n \ln n - O(n)$ on the number of steps where we choose a selected node. This again leads to a bound of $n \ln n - O(n)$, since the probability of choosing a selected node equals $1/4 - O(n^{-1/4})$. In the second phase the probability that a node on the chosen path is a free-rider equals 1/2 (independently for all the nodes). Hence, the expected time to color the path uncolored after phase 1 is bounded below by $(1/2)n \log n - O(n)$.

Theorem 2: The expected optimization time of the (1+1)GA on H-IFF is bounded below by $n \ln n + (1/2)n \log n - O(n) \approx 1.193\, n \log n$.

An upper bound of $O(n \log^2 n)$ is easy to obtain ([9]) if one applies the coupon collector's theorem bottom-up level by level. We can prove the asymptotically precise bound of $O(n \log n)$. For the upper bound we can ignore free-riders.

Theorem 3: The expected time of Game 1 and the expected optimization time of the (1+1)GA on H-IFF are bounded above by $2.679\, n \log n + o(n)$.

Proof: The proof is an application of the delay sequence argument introduced by Ranade (1991) for analyzing randomized routing protocols in networks.

We investigate Game 2 which is slower than Game 1 and the (1+1)GA on H-IFF.

We consider a phase of length $c(n-1)\log n$ and estimate the probability of not coloring the root within such a phase. Our aim is to choose the constant $c = c(\varepsilon)$ in such a way that the above probability is bounded by $n^{-\varepsilon}$. The next phase starts with more colored nodes and has an even better chance to color the root. Altogether, the expected number of phases is bounded above by $(1-n^{-\varepsilon})^{-1} = 1 + O(n^{-\varepsilon})$ implying an upper bound of $cn \log n + o(n)$ on the expected waiting time of Game 1.

For each path $p = (v_1, \ldots, v_k)$ from a level-1 node $v_1 = l$ to the root $v_k = r$ we consider also another game. In the beginning, all nodes are uncolored. Node v_1 is colored when it is chosen, each node v_i is colored when v_{i-1} is colored and v_i is chosen. Using the same random experiments as in Game 2, we choose nodes from the tree one by one. The path p is called t-slow when it is not totally colored in t steps of the new game.

The crucial observation is the following one. If in Game 2 the root is not colored in t steps, then there is a path which has all properties of a t-slow path. We construct this path starting at the root which is not colored by assumption. As long as the considered node has an uncolored child we choose such a child. Afterwards, we choose the child which has been colored *later* than its sibling. On this path, we can at first color only the level-1 node. Whenever a node gets colored, its sibling has been colored before and we can color the parent in the next step. This path is called t-slow in the following. Now we consider a phase of length $t(n) = c(n-1)\log n$ starting with a tree where all inner nodes are uncolored. Then

$$\begin{aligned}
&\text{Prob(root } r \text{ is not colored at the end of the phase)}\\
&= \text{Prob}(\exists \text{ level-1 node } v \mid \text{ path } v \to r \text{ is } t(n)\text{-slow})\\
&\leq (n/2)\text{Prob(a path with } \log n \text{ nodes is } t(n)\text{-slow)}.
\end{aligned}$$

The last inequality follows from the fact that we have $n/2$ level-1 nodes and the fact that all paths have the same length $\log n$.

A path is $t(n)$-slow if we do not color all its $k = \log n$ nodes. Since in each step we have a probability of exactly $1/(n-1)$ to color one node on the path, this is the scenario of a Chernoff bound with $t(n)$ trials and success probability $1/(n-1)$. Let X be binomially distributed with these parameters. Then, since $E(X) = c \cdot \log n = ck$, by Chernoff's inequality, for $c > 1$

$$\begin{aligned}
&\text{Prob(a path with } k \text{ nodes is } t(n)\text{-slow)}\\
&= \text{Prob}(X < k)\\
&= \text{Prob}\left(X < \left(1 - \left(1 - \frac{1}{c}\right)\right) \cdot E(X)\right)\\
&\leq \left(\frac{e^{-(1-1/c)}}{(1-(1-1/c))^{(1-(1-1/c))}}\right)^{E(X)}\\
&= \left(c^{1/c} \cdot e^{(1-1/c)}\right)^{c \cdot \log n}\\
&= \left(c \cdot e^{c-1}\right)^{\log n}.
\end{aligned}$$

If this probability is bounded above by $n^{-(1+\varepsilon)}$ we obtain the result that the probability of not coloring the root within one phase is bounded above by $(n/2)n^{-(1+\varepsilon)} \le n^{-\varepsilon}$ and, as described above, this proves the theorem. Hence, it is sufficient to choose c such that

$$c \cdot e^{c-1} = \left(\frac{1}{2}\right)^{1+\varepsilon}.$$

Choosing $c = 2.679$ this condition is fulfilled for some $\varepsilon > 0$. ■

Our bounds are sufficiently tight to explain the efficiency of the (1+1)GA on H-IFF functions. Moreover, we have seen that the optimization process essentially consists of two phases. The first phase has the task to color all level-1 nodes or equivalently to create all building blocks of length 2. This seems to be an easy task. However, it takes already $\Theta(n \log n)$ steps. In this phase, we also create many other much longer building blocks. The second phase is essentially equivalent to the coloring of one path from a level-1 node to the root. Altogether, our analysis also provides much insight how the (1+1)GA works.

The results are also sufficient to estimate the expected time until a node v on level l is colored in Game 1 or until a building block of length 2^l at a special position is created. The expected number of steps with cut positions in the subtree rooted at v equals $\Theta(2^l \cdot l)$ (Theorem 2 and Theorem 3 for $n = 2^l$). The probability that a cut position is within this tree equals $(2^l - 1)/(n-1)$. Thus we obtain an expected time of $\Theta(2^l \cdot l \cdot (n-1)/(2^l - 1)) = \Theta(n \cdot l)$ until a node on level l is colored.

Our bounds are tight up to small multiplicative constants and small additive terms of smaller order. Nevertheless, these error terms may be essential for small values of n. Therefore, it makes sense to consider experimental results.

VI. Experimental Results

The (1+1)GA is run on H-IFF$_n$, $n = 2^k$, $k \in \{6, \dots, 12\}$. For each n, we have performed 100 runs and the optimization time of the (1+1)GA is divided by $n \log n$. The results (best run, worst run, average) are contained in Table I.

TABLE I
Normalized optimization time of the (1+1)GA on H-IFF$_n$ (number of steps divided by $n \log n$).

n	best	worst	average
64	0.55	2.04	1.15
128	0.66	2.35	1.20
256	0.79	2.34	1.33
512	0.79	2.10	1.37
1024	0.97	1.97	1.41
2048	0.99	2.01	1.45
4096	1.10	2.15	1.47

These results can be explained as follows. The best case is for small n better than the leading term of the lower bound for the average time. This happens because of the variance of the stochastic process and because the lower bound also contains negative linear terms. The coupon collector's theorem for the first phase gives a result which is close to

$$n \ln(n/4) = n \ln n - (2 \ln 2)n = n \ln n - 1.386\, n.$$

Also the lower bound $(n-1)(\log n - 1)$ contains a negative linear term. These linear terms are important for small n. This also explains why the values in the column "average" are increasing. Nevertheless, they are much closer to the leading constant of the lower bound than to the leading constant of the upper bound. We believe that this is also true for very large n. In the proof of the upper bound we estimate the probability of a union of events by the sum of the probabilities. This seems to be a rough estimate implying that the exact bound even for Game 2 is smaller than the upper bound. Moreover, the (1+1)GA is faster than Game 2, since Game 2 ignores free-riders.

All in all, our asymptotic results fit the experimental results already for small n.

VII. Conclusion

H-IFF$_n$ is a hierarchically decomposed function which looks "crossover-friendly". A recombinative hill-climber called (1+1)GA which ensures diversity by adding $\bar{x}$ to a population containing x is investigated. The dynamics cannot be analyzed exactly enough to give much insight. An asymptotically tight estimation of the expected optimization time illustrates how the optimization process works. Moreover, the analysis introduces the important method of delay sequences into the analysis of evolutionary algorithms.

References

[1] J. Culberson, "Genetic invariance: A new paradigm for genetic algorithm design," Tech. Rep. TR92-02, University of Alberta, 1992.

[2] C. Höhn and C. Reeves, "Are long path problems hard for genetic algorithms?," in *Proceedings of the 4th Conference on Parallel Problem Solving from Nature,* H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds., Berlin: Springer-Verlag 1996, vol. 1141 of LNCS, pp. 134-143.

[3] T. Jansen and I. Wegener,"Real royal road functions – where crossover is provably essential" in *Proceedings of Genetic and Evolutionary Computation Conference 2001,* Lee Spector et al., Ed., San Mateo, CA: Morgan Kaufmann, 2001, pp. 375–382.

[4] R. Motwani and P. Raghavan, *Randomized Algorithms.* Cambridge, U.K.: Cambridge Univ. Press, 1995.

[5] B. Naudts and J. Naudts, "The effect of spin-flip symmetry on the performance of the simple GA," in *Proceedings of the 5th Conference on Parallel Problem Solving from Nature,* A. E. Eiben et al., Ed., Berlin: Springer-Verlag, 1998, vol. 1498 of LNCS, pp. 67-76.

[6] A. G. Ranade, "How to emulate shared memory," *Journal of Computer and System Sciences,* vol. 42, pp. 307–326, 1991.

[7] C. Van Hoyweghen, D. E. Goldberg, and B. Naudts, "Building block superiority, multimodality and synchronization problems," in *Proceedings of the Genetic and Evolutionary Computation Conference 2001,* Lee Spector et al., Ed., San Mateo, CA: Morgan Kaufmann, 2001, pp. 694–701.

[8] C. Van Hoyweghen, B. Naudts, and D. Goldberg, "Spin-flip symmetry and synchronization," Submitted to the journal of Evolutionary Computation, 2001.

[9] R. Watson, "Analysis of recombinative algorithms on a non-separable building block problem," in *Foundations of Genetic Algorithms 6,* W. Martin and W. Spears, Eds., San Mateo, CA: Morgan Kaufmann, 2001, pp. 69-89.

[10] R. Watson, G. S. Hornby, and J. B. Pollack, "Modeling building-block interdependency," in *Proceedings of the 5th Conference on Parallel Problem Solving from Nature,* A. E. Eiben et al., Ed., Berlin: Springer-Verlag, 1998, pp. 97–106.