Crossover: The Divine Afflatus in Search

David Iclănzan
Department of Computer Science
Babeş-Bolyai University, Kogălniceanu no. 1
Cluj-Napoca, 400084, Romania
david.iclanzan@gmail.com

ABSTRACT

The traditional GA theory is pillared on the Building Block Hypothesis (BBH) which states that Genetic Algorithms (GAs) work by discovering, emphasizing and recombining low order schemata in high-quality strings, in a strongly parallel manner. Historically, attempts to capture the topological fitness landscape features which exemplify this intuitively straight-forward process, have been mostly unsuccessful. Population-based recombinative methods had been repeatedly outperformed on the special designed abstract test suites, by different variants of mutation-based algorithms. Departing from the BBH, in this paper we seek to exemplify the utility of crossover from a different point of view, emphasizing the creative potential of the crossover operator. We design a special class of abstract test suites, called Trident functions, which exploits the ability of modern GAs to mix good but *significantly different* solutions. This approach has been so far neglected as it is widely believed that disruption caused by mating individuals that are too dissimilar is harmful. We anticipate that hybridizing different designs induces a complex neighborhood structure unattainable by trajectory-based methods which can conceal novel solutions. Empirical results confirm that the proposed class of problems can be solved efficiently only by population-based panmictic recombinative methods, employing diversity maintaining mechanisms.

Categories and Subject Descriptors

F.2 [**Analysis of algorithms and problem complexity**]: General

General Terms

Algorithms, Theory, Experimentation, Performance

Keywords

Fitness Landscape, Crossover, Hybridization, Creativity

GECCO'07, July 7–11, 2007, London, England, United Kingdom.
Copyright 2007 ACM 978-1-59593-698-1/07/0007 ...$5.00.

1. INTRODUCTION

Evolutionary Algorithms (EAs) are inspired and closely linked to Darwinian evolution. The complex schemes involved in the transmission of biological information make these methods suited for changing environments. Recent results had shown that in dynamical environments induced by exogenous noise, recombinative methods clearly outperform hill-climbers [5, 17]. However, as applications of EAs consider mostly fixed fitness landscapes, a major criticism of EAs is that biological metaphors may be unnecessarily complex.

A major open problem regarding the fundamentals of GAs is to characterize the topological features of *static* fitness landscapes for which crossover is an efficient operator. These landscapes would exemplify the utility of the nature inspired algorithms by revealing the class of problems for which EAs are most suited, outperforming other optimizers such as hill-climbers. Despite the major work in this field, it is still unclear how an EA explores a search space.

According to the intuition behind the BBH [7, 3], GAs should perform well when they are able to discover above-average-fitness low-order schemata and recombine them to produce even more fit higher-order schemata in a recursive manner. So far, attempts to design abstract test functions suited for this hierarchical, straight-forward process have been unsuccessful. Population-based recombinative methods had been repeatedly outperformed on the special designed abstract test suites, by different variants of mutation-based algorithms.

Recent developments [9] had shown that a certain hill-climber operating over the building-block space is able to solve problems even with epistatic and unfavorable genetic linkage.

Test problems for GAs were usually developed under the intuitions of the BBH. As it is believed that crossover should produce successful offspring on average, test problems were usually devised accordingly.

Leaving aside the suggestions of the BBH, the objective of this paper is to develop test suites that exploit the generative potential of the crossover operator.

The creative potential of the crossover operator had been already conjectured many times [20, 1, 18]. A great asset of EAs is the ability to simultaneously process and combine many *different* designs. The promoting idea behind recombination, as it was originally proposed [7], is to combine different sub-solutions from varying individuals. Nevertheless, there is widely spread methodology which emphasize another aspect of the recombination, namely that it preserves the similarity exhibited by the parents. This idea is supported in classical approaches, where niching and speciation methods restrict mating to individuals that are genotypically similar. The same view manifests in spatially distributed GAs [19] where breeding is promoted within local populations. These approaches advocate the idea that parents selected from two different fitness peaks are likely to produce an offspring that lands in the "valley" between.

But what if valuable solutions lay between two different fitness peaks? This may be the case in many design problems, where hybridizing two prototypes can result in a completely new, valuable solution. In these cases, though crossover may not produce successful offspring on average, it conceals a great generative ability, revealing novel designs which would be unattainable otherwise.

We think that these functions may exemplify the utility of EAs in static environments. The points from the complex neighborhood structure defined by crossover on two valuable but different solutions, are expected to be intractable by trajectory methods, provided that these points do not have a large basin of attraction. In this case the problem can be solved only by employing the generative potential of the recombination operator.

The remaining of the paper is organized as follows: in the next section we review the history of the quest for "GA friendly" test functions. In Section 3 the class of functions which emphasize the creative potential of GAs (Trident functions) is introduced. Experimental setup and results on these test suites are presented in Section 4. Section 5 concludes.

2. HISTORICAL BACKGROUND

The Royal Road (RR) problems [2] were an early attempt to capture the problem structure, which enables GAs to outperform other methods. The adopted strategy was to find a set of features that are of particular relevance to GAs and test the performance of these algorithms on landscapes containing those features. It was recognized that major tenets behind the BBH are the notion of problem decomposition and the assembly of solutions from sub-solutions. Subsequently, a set of functions were constructed, which clearly emphasized a gross-scale building-block structure with low-order building-blocks that recombine to higher-order ones. The functions were expected to lay out a "royal road" for GAs and the intention was to exemplify the class of problems that required the recombinative aspects of the GA. However, a Random Mutation Hill-Climber (RMHC), which accepts states with equal objective function value, greatly outperformed GAs on these test suites.

The RR functions can be solved by mutation-based hill-climbers much faster than by GAs, because they exhibit separable building-block structure with tight linkage and do not contain any local optima. Even when deception is introduced and each building-block corresponds to a deceptive trap function, the Macro Mutation Hill-Climber (MMHC) can outperform the GA, by benefiting from the tight linkage and cluster mutating one sub-function at a time [10].

In order to break hill-climbers, problems with unfavorable epistatic linkage and unfavorable genetic linkage were devised. The former refers to building-block interdependency (non-separability), while the latter corresponds to building-blocks formed by non-adjacent alleles. Watson et al. [22] introduced a class of hierarchically decomposable functions, which presented strong non-linear hierarchical building-block interdependency. In a later development, Pelikan et al. [15] proposed the hierarchical trap function, where a basic deceptive function is used on each level to make the problem even harder.

On problems with poor genetic linkage, classical crossover is not able to combine good building-blocks effectively. Thus, considerable effort was invested at developing algorithms that discover and represent the interdependency of alleles, resulting in more intelligent crossover operators [6, 12, 11, 16, 14, 23]. As a result, competent methods [4] can address problems which present both epistatic and unfavorable genetic linkage. It was considered that this class of functions exemplifies those problems for which GAs are well-suited.

Hill-climbing in building-block space was proposed and shown to be more efficient than selectorecombinative GAs on deterministic additively-separable problems of bounded difficulty [17]. But a recent development [9] showed that a certain hill-climber operating over the building-block space can solve problems with epistatic and unfavorable genetic linkage. The new Building Block Hill-Climber (BBHC) is able to solve hierarchical problems with random linkages. The method uses hill-climb search experience to learn the problem structure. The neighborhood structure is adapted whenever new knowledge about the underlaying BB structure is incorporated into the search. The BBHC holds a major qualitative advantage, scaling as $O(l \cdot log(l))$ compared to $O(l^{1.5} \cdot log(l)$, which is at least required by population-based recombinative methods [14].

The new result suggests that if a problem has a nice structure, even if "hidden" like the building-block space with unfavorable genetic linkage, a proper hill-climber can outperform population-based recombinative methods, *without requiring extra domain knowledge.*

We suspect that the idea of a GA marching on a fitness landscape is maybe a little bit romantic; a suitable hill-climber is almost certainly quicker if there is a nice structure of the problem to be exploited. Maybe we should look for hard problems, which can be solved somewhat slothfully by GAs, but are intractable using other methods.

In the next chapter such a function is introduced, for which we hope GAs are well-fitted.

3. HYBRIDIZATION OF DIFFERENCES

The reason why hill-climbers outperform population-based recombinative methods on the test suites proposed so far, may be their non-deceptive nature at *some* level. Deception may manifest in more then one form [8], but the general idea is that "a deceptive function is one in which low-order schema fitness averages favor a particular local optimum, but the global optimum is located at that optimums complement" [3]. The RR function does not contain any local optima; hierarchical problems are fully deceptive in the Hamming space but are fully non-deceptive in the building-block or crossover space. Recent local-search literature authors have emphasized the importance of using a good neighborhood operator [21]. Thus, it is not that surprising that a certain hill-climber employing a good neighborhood operator (which engenders a building-block-wise search in the case of hierarchical functions), can outperform methods which operate via populations.

We reason that in order to defeat hill-climbers, problems must contain a degree of deception, which can not be overcome by a neighborhood operator induced by one point in the search space. This of course will hinder GAs performance also, as the mutation works in the neighborhood of one individual and short-term selection may favor deceptive search paths. However, EAs posses a great asset by having a more complex neighborhood structure generated by the recombination operator, which takes into account at least two individuals. This may help the methods to escape the local optima and overcome deceptiveness.

The problem representation together with the neighborhood structure defines the search landscape. We argue that there are problems where only search landscapes transformed by crossover may be efficiently exploitable. In the followings we give an example for such a class of problems called the Trident functions (TF).

3.1 The Trident Function

TF accepts bitstrings of the length $2k$ where $k \geq 2$ and uses a function of unitation (which depends on the number of ones in a bitstring, and not on their positions) as underlaying structure:

$$base(x) = \|2 \cdot u(x) - |x|\| \tag{1}$$

where $u(x)$ is the unitary of x (the number of ones) and $|x|$ is the length of x.

The base function has its minimum in 0 which is generated by strings with an equal number of 1's and 0's: $u(x) = |x| - u(x)$. The maximum is attained by strings formed by all 1's or all 0's with a corresponding value of $|x|$.

The next component of the TF is a contribution function which rewards *certain configurations* of strings that have an equal number of 1's and 0's. Let $L = x_1, x_2, \ldots, x_{\frac{n}{2}}$ be the first half of the binary string x of length n and $R = x_{\frac{n}{2}+1}, x_{\frac{n}{2}+2}, \ldots, x_n$ the second one. Then, we define the contribution function for this paper based on the exclusive OR (XOR) relation:

$$contribution(x) = \begin{cases} 2 \cdot |x| & \text{, if } L = \bar{R}; \\ 0 & \text{, otherwise.} \end{cases} \tag{2}$$

where $\bar{R}$ stands for the bitwise negation of R.

Please note that the contribution function does not have a basin of attraction; it rewards fully an input or it does not reward it at all. Finding the maxima of such a function is equivalent to the *needle in the haystack* problem. As there are no better search methods for this class of function than the random-search, these function are also resistant to biased mutation-based search.

The TF is defined as the sum of the base and the contribution function:

$$trident(x) = base(x) + contribution(x) \tag{3}$$

Figure 1 presents the graphical interpretation of the Trident function.

TF has its maximum in the points rewarded by the contribution function. Here it takes the value $2 \cdot |x|$ as the base function in these points attends the minimum 0.

TF is very hard for mutation-based algorithms because the base function leads away the search from the region where global optima lay. Even if a random state is generated with equal number of 1's and 0's, it is very unlikely for large problem instances that the contribution function will reward that string. If the algorithm does a biased search, it will be immediately drawn away from the minimum of the base function, towards regions with higher base function fitness.

The TF can defeat macromutation hill-climbers also, as local and global optima are very distant in the Hamming space. The chance of jumping from local optima to a global one is minimal as $\frac{n}{2}$ bits must be changed simultaneously.

Also, there are no "hidden" structures which could be easily exploited. The "building-blocks" L and R are rewarded if and only if their context i.e. the counterpart half of the string is compatible. As TFs have $2^{\frac{n}{2}}$ global solutions, the probability of this happening for randomly generated strings is $P_{hit} = \frac{2^{\frac{n}{2}}}{2^n} = \frac{1}{2^{\frac{n}{2}}}$.

What about GAs? Global optima can be found quite easily if the GA is mixing good but different solutions. Let us take the example where $n = 8$ and we have two strings at each local optimum: $s_1 = 00000000$ and $s_2 = 11111111$. The one-point crossover between s_1 and s_2 will produce the optimal strings $s_3 = 00001111$ and $s_4 = 11110000$ with the probability $P = \frac{1}{n-1} = \frac{1}{7}$. When using two-point crossover, we have $\frac{n}{2} - 1 = 3$ favorable cases. The favorable crossing points pairs are $\{(1,7), (2,6), (3,5)\}$. Optimal strings may not result only from the breeding of individuals located at local optima. For example, the one-point crossover between $s_5 = 00100001$ and $s_6 = 11111101$ between loci 4 and 5 will also produce an optimal solution $s_7 = 00101101$. The important aspect is to combine *different* candidate solutions.

The TF is the abstracted form of problems where several highly different good solutions exist, and hybridizing these solutions *may* result in a completely new, valuable design. Even if crossover does not produce above average individuals on a regular basis, it may create occasionally an exceptional organism. Thus, crossover has a generative potential which we believe should not be neglected by restricting the recombination to genotypically similar individuals.

3.2 Natural Metaphor

Hybridization between related species is a common occurrence in nature and it plays an important role in the evolution of some species. Hybrids may have a beneficial combination of traits, allowing them to succeed in a marginal habitat where the two parent species are disadvantaged. If the hybrid is successful, new species may emerge within a relatively small number of generations. This leads to the hypothesis that life is a genetic continuum rather than a series of self-contained species.

Unlike mutation, which affects only one gene, hybridization simultaneously affects multiple variations across genes or gene combinations. This translates to big jumps on the fitness landscape. However, these jumps are not random as in the case of the macromutation. Crossover may mix certain well established features which may not be that important for the parents, but their combination is spectacular and it is highly rewarded in the resulting hybrid. Hybrids in nature are often stronger than either parent variety. This phenomenon is known as *hybrid vigor* or *heterosis*.

In the next section we analyze how different variants of hill-climbers and GAs perform on the proposed class of functions.

4. RESULTS

We tested the RMHC, the MMHC, the Simple Genetic Algorithm (SGA) and a special genetic algorithm which em-

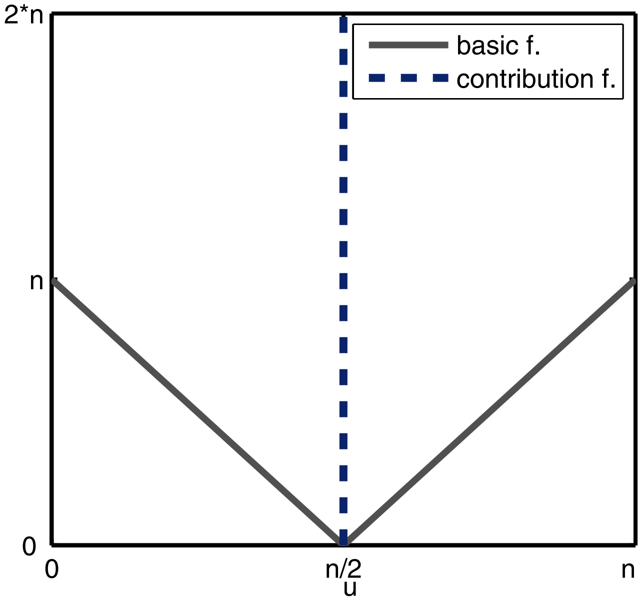

Figure 1: The Trident Function, where u is the unitary of the input string. The base function is deceptive, leading away the search from the area which contains global optima. The contribution function has no basin of attraction so its maximum is very hard to detect. Note that the contribution function does not reward all strings with $u = \frac{n}{2}$; only special configurations are rated.

ploys diversity maintenance, called Deterministic Crowding (DC) on the 16, 32, 64 respectively 128-bit versions of the TF. A number of 100 independent runs were averaged. The algorithms ran until they found a global optimum or 10^6 function evaluations were exceeded.

The difficulty of the TF grows exponentially with the problem size. The 16-bit version of the problem is utterly easy as global optimum occupies $\frac{1}{256}$-th of the search space. A simple random sampling of the search space will yield an optimal solution in 256 steps on average. Anyhow, increasing the problem size to 64-bit will drop the probability of randomly hitting an optimal solution to less then 1 to 4.e+9. We were especially interested how the methods can handle the exponential explosion and how the reduction of the fraction occupied by global optima in report with the search space size, will affect their scalability.

The methods and the settings used for each algorithm are briefly described as follows.

4.1 Random Mutation Hill-Climber

RMHC [2] chooses at each iteration a locus at random to mutate. If the mutation leads to an equal or higher fitness, then the new state is accepted.

The power of the RMHC lays in its ability to travel on fitness neutral regions of the search space, explaining its success on the RR functions. Big plateaus with equal fitness are not a characteristics of the TFs, therefore RMHC will behave similarly to the Next Ascent Hill-Climber.

A random restart was applied in our test, if the RMHC could not improve for 10^4 consecutive evaluations.

```
DC

  1. Initialize the population randomly;

  2. Pick two parents, $p_1$ and $p_2$ at random from the
     population;

  3. Do crossover and mutation to produce a pair of
     offsprings $c_1$ and $c_2$;

  4. Pair-up each offspring with one parrent
     according to the pairing rule below;

  5. For each parent-offspring pair, if the
     offspring is better than the parent then
     replace the parent with the offspring;

  6. If termination condition not met goto 2;

Pairing rule:
if $H(p_1,c_1) + H(p_2,c_2) < H(p_1,c_2) + H(p_2,c_1)$ then
pair $p_1$ with $c_1$, and $p_2$ with $c_2$, otherwise pair $p_1$
with $c_2$, and $p_2$ with $c_1$, where H gives the
genotypic Hamming distance between two
individuals.
```

Figure 2: Deterministic Crowding as used in the experiments.

4.2 Macro Mutation Hill-Climber

MMHC had been shown to be a very powerful hill-climbing method, which can outperform GAs even on problems where each building-block corresponds to a deceptive trap function, provided that the problem has a tight linkage [10].

In the mutation operator of the MMHC, the genome is viewed as a string. Two distinct points on the ring are randomly selected and the loci in the smaller section are set to random alleles.

In our tests, if the MMHC could not improve for 10^5 epochs, it was restarted from a random state, in the limit of the 10^6 function evaluations.

4.3 Simple Genetic Algorithm

The SGA [7, 3] is a GA that uses fitness-proportionate selection, one-point crossover, and point mutation to evolve a single population of bit strings, with each generation completely replacing the previous one. The population size was set to $pop\_size = 500$, the mutation rate to $p_{mut} = 0.005$ and the crossover rate was $c = 0.8$.

4.4 Deterministic Crowding

Deterministic Crowding operates on the premise of restricted competition rather than restricted mating. The population is panmictic so genotypically different individuals may be recombined. This accommodates well the view on the importance of hybridization. Diversity in maintained by restricting the competition to parents and their own offspring. In this way, subpopulations that are occupying different niches do not need to outperform each other in order to propagate. The outline of the DC is presented in Figure 2. For a description in detail of the method, see [13].

For the DC we used the same parameter settings as for the SGA.

Table 1: Performance of the Random Mutation Hill-Climber (RMHC), Macro Mutation Hill-Climber (MMHC), Simple Genetic Algorithm (SGA) and the Deterministic Crowding (DC) on the TFs. Column "Succ. rate" contains the number of successful runs where the methods find global optima. Column "Avg. nr." contains the number of average function evaluations needed to find the optima and "Max. nr." the maximum number of evaluations needed when all runs were successful. In the case of the population-based methods, column "Nr. opt." contains the average number of different optima within the correctly converged population.

TF size	**16**				**32**			
	Succ. rate	Avg. nr.	Max. nr	Nr. opt.	Succ. rate	Avg. nr.	Max. nr.	Nr. opt.
RMHC	**85%**	400020	-	/	**3%**	337136	-	/
MMHC	**100%**	4213	33528	/	**37%**	471018	-	/
SGA	**100%**	241	1390	1.46	**25%**	8435	-	1.56
DC	**100%**	250	2111	23.94	**100%**	15771	23883	6.69

TF size	**64**				**128**			
	Succ. rate	Avg. nr.	Max. nr	Nr. opt.	Succ. rate	Avg. nr.	Max. nr.	Nr. opt.
DC	**100%**	46816	61366	6.02	**100**	112557	157890	4.32

4.5 Numerical Results

The numerical results of the experiments are summarized in Table 1. On the 64 and 128-bit versions of the TF, only the results of the DC are reported as the hill-climbers and the SGA failed in all runs on these suites.

As expected, the worst behavior on the TFs was shown by the RMHC. Even for the very easy 16-bit version of the problem, the success rate is only 85%. Similar to the other hill-climber, the MMHC, solutions are found at very high cost and only due to the random restart mechanism. The number of function evaluations required to identify optima, exceeded by orders of magnitude the amount that would be required by random-search. As the problem size increases, becoming unaddressable by random sampling, the hill-climbers fail in all runs due the deceptive nature of the TF.

When SGA succeeds, its performance is the fastest, being much better then it would be required by random-search. This shows that even simple recombinative algorithms have the potential to exploit the features of the TF landscape. However, as the problem size increases, the SGA fails to find optima due to lack of improper initial sampling. Without the proper identification of the valuable regions from the search space, the population is quickly shifted towards the basin of attraction of a single local optimum. A significant increase of the population size could address this problem but then success would come at high costs.

The only competent algorithm on the TFs was the DC. It succeeded in absolutely all runs, being able to identify global optima within a maximum of 16% of the allowed function evaluations. In all cases, several optimum points were detected. The success of the algorithm derives from its diversity maintaining mechanism combined with the panmictic population.

We once again emphasize the importance of the capability to mix different designs; only then recombination can become creative. An algorithm with diversity maintaining mechanism, but with crossover restricted to similar individuals, would also fail on the TFs.

5. CONCLUSIONS

This paper has examined some questions concerning the fundamentals of EAs. The shortcomings of existing building-block style test functions were surveyed and a view which promotes and emphasizes the generative potential of EAs was presented. According to this conception, the great asset of EAs lays in their capability to concomitantly evolve and hybridize different designs.

In the light of this concept, a new class of test problems, called Trident functions (TFs) was introduced. The TFs are dominated by a fully deceptive base function as global optima coincide with the minima of this function. The discrete optimal solutions are defined by a contribution function which rewards points from the search space where certain different genotypical features appear concomitantly. As the contribution function does not have an attractor basin, the deceptiveness of the base function can not be overcome using only simple neighborhood structures.

Ancient people believed that the gods blew on creative people, who would then inhale the god's breath and have a brilliant idea. At a first sight TF seems so hard that one would think that there is a need for "divine inspiration" in order to efficiently locate its optima. However, the TF is only the abstracted form of certain design problems where several good, locally optimal drafts are easy to find, dominating the search space (deception) and the real good designs result from the hybridization of different drafts. Furthermore, the complex layouts defining the best solution only emerge in "reactive regions" where the correct particular features appear simultaneously; there is no sequence of improving designs to these solutions (needle in the haystack). Nevertheless, crossover possesses the creative potential i.e the more complex neighborhood structure, which enables it to identify these solutions by mixing features from different drafts, until the correct configuration is detected.

The experimental results confirm that the proposed class of functions contains topological features, which can be efficiently exploited only by recombinative methods mixing different good solutions. It seems that if the population of a GA is allowed to genotypically converge, its behavior becomes less distinct from the workings of trajectory-based methods.

Similarly to the development of other test problems, the main purpose of this work has been to reveal and provide a better understanding of the essential mechanisms in EAs. The results presented suggest that population-based recombinative methods provide fundamentally different advantages compared to non-recombinative algorithms. Hill-climbers may very quickly find good solutions on relatively simple landscapes while population-based recombinative methods can tackle more complex problems, where global optima only arise from combination of different features.

6. ACKNOWLEDGMENTS

This work was sponsored by MEdC-CNCSIS under grant AT-8/70-2006 and by the Sapientia Institute for Research Programs (KPI).

7. REFERENCES

[1] S. Chen. *Is the Common Good? A New Perspective Developed in Genetic Algorithms*. PhD thesis, Robotics Institute, Carnegie Mellon University, 1999.

[2] S. Forrest and M. Mitchell. What makes a problem hard for a genetic algorithm? some anomalous results and their explanation. *MACHLEARN: Machine Learning*, 13, 1993.

[3] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.

[4] D. E. Goldberg. *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[5] D. E. Goldberg, K. Sastry, and X. Llora;. Toward routine billion-variable optimization using genetic algorithms: Short communication. *Complex.*, 12(3):27–29, 2007.

[6] G. R. Harik and D. E. Goldberg. Learning linkage. In R. K. Belew and M. D. Vose, editors, *FOGA*, pages 247–262. Morgan Kaufmann, 1996.

[7] J. H. Holland. *Adaptation in natural artificial systems*. University of Michigan Press, Ann Arbor, 1975.

[8] J. Horn and D. E. Goldberg. Genetic algorithm difficulty and the modality of fitness landscapes. In L. D. Whitley and M. D. Vose, editors, *FOGA*, pages 243–269. Morgan Kaufmann, 1994.

[9] D. Iclanzan and D. Dumitrescu. Overcoming Hierarchical Difficulty by Hill-Climbing the Building Block Structure. In *GECCO '07: Proc. of the Genetic and Evolutionary Computation Conference*, July 2007. (accepted).

[10] T. Jones. *Evolutionary Algorithms, Fitness Landscapes and Search*. PhD thesis, University of New Mexico, Albuquerque, NM, 1995.

[11] H. Kargupta. Gene expression: The missing link in evolutionary computation. In *Genetic Algorithms in Engineering and Computer Science*, pages 159–160, Las Vegas, Nevada, USA, 8 July 2000.

[12] P. Larranaga and J. A. Lozano. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, Norwell, MA, USA, 2001.

[13] S. W. Mahfoud. *Niching methods for genetic algorithms*. PhD thesis, Dept. General Engineering, University of Illinois, Urbana, Illinois, 1995. Also, IlliGAL Report No. 95001.

[14] M. Pelikan. *Hierarchical Bayesian optimization algorithm: Toward a new generation of evolutionary algorithms*. Springer Verlag, 2005.

[15] M. Pelikan and D. E. Goldberg. Escaping hierarchical traps with competent genetic algorithms. In L. S. et al., editor, *GECCO '01: Proc. of the Genetic and Evolutionary Computation Conference*, pages 511–518, San Francisco, California, USA, 7-11 2001. Morgan Kaufmann.

[16] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz. BOA: The Bayesian optimization algorithm. In W. B. et al., editor, *GECCO '99: Proceedings of the Genetic and Evolutionary Computation Conference*, volume I, pages 525–532, Orlando, FL, 13-17 July 1999. Morgan Kaufmann Publishers, San Fransisco, CA.

[17] K. Sastry and D. E. Goldberg. Let's get ready to rumble: Crossover versus mutation head to head. In *GECCO '04: Proc. of the Genetic and Evolutionary Computation Conference, Seattle, WA, USA,*, pages 126–137. Springer, LNCS, vol. 3103, June 26–30 2004.

[18] R. Shipman, M. Shackleton, and I. Harvey. The use of neutral genotype-phenotype mappings for improved evolutionary search. *BT Technology Journal*, 18(4):103–111, 2000.

[19] T. Starkweather, K. Mathias, and D. Whitley. Optimization using distributed genetic algorithms. In *Proceedings of the First International Conference on Parallel Problem Solving from Nature*, pages 176–186. Springer, 1991.

[20] G. Syswerda. Uniform crossover in genetic algorithms. In *Proceedings oThird International Conference on Genetic Algorithms*, 1989.

[21] Watson, Beck, Howe, and Whitley. Problem difficulty for tabu search in job-shop scheduling. *AIJ: Artificial Intelligence*, 143, 2003.

[22] R. A. Watson, G. Hornby, and J. B. Pollack. Modeling building-block interdependency. In *PPSN V: Proc. of the 5th International Conference on Parallel Problem Solving from Nature*, pages 97–108, London, UK, 1998. Springer, LNCS.

[23] T.-L. Yu and D. E. Goldberg. Conquering hierarchical difficulty by explicit chunking: substructural chromosome compression. In *GECCO '06: Proc. of the Genetic and Evolutionary Computation Conference*, pages 1385–1392, NY, USA, 2006. ACM Press.