Explaining Adaptation in Genetic Algorithms With Uniform Crossover: The Hyperclimbing Hypothesis

Keki M. Burjorjee
Zynga Inc.
650 Townsend Street
San Francisco, CA 94103
kekib@cs.brandeis.edu

ABSTRACT

The *hyperclimbing hypothesis* is a hypothetical explanation for adaptation in genetic algorithms with uniform crossover (UGAs). Hyperclimbing is an intuitive, general-purpose, non-local search heuristic applicable to discrete product spaces with rugged or stochastic cost functions. The strength of this heuristic lies in its insusceptibility to local optima when the cost function is deterministic, and its tolerance for noise when the cost function is stochastic. Hyperclimbing works by decimating a search space, i.e. by iteratively fixing the values of small numbers of variables. The hyperclimbing hypothesis holds that UGAs work by implementing *efficient* hyperclimbing. Proof of concept for this hypothesis comes from the use of a novel analytic technique involving the exploitation of algorithmic symmetry. We have also obtained experimental results that show that a simple tweak inspired by the hyperclimbing hypothesis dramatically improves the performance of a UGA on large, random instances of MAX-3SAT and the Sherrington Kirkpatrick Spin Glasses problem.

Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: Artificial Intelligence—*Problem Solving, Control Methods, and Search*; F.2 [**Theory of Computation**]: Analysis of Algorithms And Problem Complexity—*Miscellaneous*

General Terms

Algorithms, Theory

Keywords

Genetic Algorithms, Uniform Crossover, Hyperclimbing Hypothesis, MAXSAT, Spin Glasses

1. INTRODUCTION

Over several decades of use in diverse scientific and engineering fields, evolutionary optimization has acquired a reputation for being a kind of universal acid—a general purpose approach that routinely procures useful solutions to optimization problems with rugged, dynamic, and stochastic cost functions over search spaces consisting of strings, vectors, trees, and instances of other kinds of data structures [6]. Remarkably, the means by which evolutionary algorithms work is still the subject of much debate. An abiding mystery of the field is the widely observed utility of genetic algorithms with uniform crossover [15, 13, 12, 8]. The use of uniform crossover [1, 15] in genetic algorithms causes genetic loci to be unlinked, i.e. recombine freely. It is generally acknowledged that the adaptive capacity of genetic algorithms with this kind of crossover cannot be explained within the rubric of the *building block hypothesis*, the reigning explanation for adaptation in genetic algorithms with strong linkage between loci [7]. Yet, no alternate, scientifically rigorous explanation for adaptation in genetic algorithms with uniform crossover (UGAs) has been proposed. The *hyperclimbing hypothesis*, presented in this paper, addresses this gap. This hypothesis holds that UGAs perform adaptation by implicitly and efficiently implementing a global search heuristic called *hyperclimbing*.

GECCO'12, July 7-11, 2012, Philadelphia, Pennsylvania, USA.
ACM 978-1-4503-1177-9/12/07.

2. THE HYPERCLIMBING HEURISTIC

For a sketch of the workings of a hyperclimbing heuristic, consider a search space $S = \{0,1\}^\ell$, and a (possibly stochastic) fitness function that maps points in S to real values. Let us define the *order* of a schema partition [11] to simply be the order of the schemata that comprise the partition. Clearly then, schema partitions of lower order are coarser than schema partitions of higher order. The *effect* of a schema partition is defined to be the variance of the expected fitness of the constituent schemata under sampling from the uniform distribution over each schema. So for example, the effect of the schema partition $\#{*}{*}\#{*}{*} = \{0{*}{*}0{*}{*},\ 0{*}{*}1{*}{*},\ 1{*}{*}0{*}{*},\ 1{*}{*}1{*}{*}\}$ is

$$\frac{1}{4}\sum_{i=0}^{1}\sum_{j=0}^{1}(F(i{*}{*}j{*}{*}) - F({*}{*}{*}{*}{*}{*}))^2$$

where the operator F gives the expected fitness of a schema under sampling from the uniform distribution. A hyperclimbing heuristic starts by sampling from the uniform distribution over the entire search space. It subsequently identifies a coarse schema partition with a non-zero effect, and limits future sampling to a schema in this partition with above average expected fitness. In other words the hyperclimbing heuristic fixes the defining bits [11] of this schema in the population. This schema constitutes a new (smaller) search space to which the hyperclimbing heuristic is recursively applied. Crucially, the act of fixing defining bits in a population has the potential to "generate" a detectable non-zero effect in a schema partition that previously had a negligible effect. For example, the schema partition $*\#***\#$ might have a negligible effect, while the schema partition $1\#*0*\#$ has a detectable non-zero effect.

At each step in its progression, hyperclimbing is sensitive, not to the fitness value of any individual point, but to the sampling means of relatively coarse schemata. This heuristic is, therefore, *natively* able to tackle optimization problems with stochastic cost functions. Considering the intuitive simplicity of hyperclimbing, this heuristic has almost certainly been toyed with by other researchers in the general field of discrete global optimization. In all likelihood it was set aside each time because of the seemingly high cost of implementation for all but the smallest of search spaces or the coarsest of schema partitions. Given a search space comprised by ℓ binary variables, there are $\binom{\ell}{o}$ schema partitions of order o. For any fixed value of o, $\binom{\ell}{o} \in \Omega(\ell^o)$ [5]. The exciting finding presented in the full version of this paper [4] and in a recent dissertation [3] is that UGAs can implement hyperclimbing cheaply for large values of ℓ, and values of o that are small, but greater than one.

3. RAMIFICATIONS

If the hyperclimbing hypothesis is sound, then the UGA is in good company. Hyperclimbing belongs to a class of heuristics that perform *global decimation*. Global decimation, it turns out, is the state of the art approach to solving large, hard instances of SAT [9]. Conventional global decimation strategies—e.g. Survey Propagation [10], Belief Propagation, Warning Propagation [2]—use message passing algorithms to obtain statistical information about the space being searched. This information is then used to fix the values of one, or a small number, of search space attributes, effectively reducing the size of the search space. The decimation strategy is then recursively applied to the smaller search space. Survey Propagation, perhaps the best known global decimation strategy, has been used along with *Walksat* [14] to solve instances of SAT with upwards of a million variables. The hyperclimbing hypothesis holds that in practice, UGAs also perform adaptation by decimating the search spaces to which they are applied. Unlike conventional decimation strategies, however, a UGA obtains statistical information about the search space *implicitly*, by means other than message passing.

Useful as it may be as an explanation for adaptation in UGAs, the ultimate value of the hyperclimbing hypothesis may lie in its *generalizability*. In [3], the notion of a unit of inheritance—i.e. a gene—was used to generalize this hypothesis to account for adaptation in simple genetic algorithms with strong linkage between chromosomal loci. It may be possible for the hyperclimbing hypothesis to be generalized further to account for adaptation in other kinds of evolutionary algorithms, In general, such algorithms may perform adaptation by efficiently identifying and progressively fixing above average "aspects"—*units of selection* in evolutionary biology speak—of the chromosomes under evolution. The precise nature of the unit of selection in each case would need to be determined.

4. REFERENCES

[1] D.H. Ackley. *A connectionist machine for genetic hillclimbing.* Kluwer Academic Publishers, 1987.

[2] Alfredo Braunstein, Marc Mézard, and Riccardo Zecchina. Survey propagation: an algorithm for satisfiability. *CoRR*, cs.CC/0212002, 2002.

[3] K.M. Burjorjee. *Generative Fixation: A Unifed Explanation for the Adaptive Capacity of Simple Recombinative Genetic Algorithms.* PhD thesis, Brandeis University, 2009.

[4] K.M. Burjorjee. Explaining adaptation in genetic algorithms with uniform crossover: The hyperclimbing hypothesis. *Arxiv preprint arXiv:1204.3436*, 2012.

[5] T. H. Cormen, C. H. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* McGraw-Hill, 1990.

[6] D. B. Fogel. *Evolutionary Computation : Towards a New Philosophy of Machine Intelligence.* IEEE press, 2006.

[7] David E. Goldberg. *The Design Of Innovation.* Kluwer Academic Publishers, 2002.

[8] Li Huifang and Li Mo. A new method of image compression based on quantum neural network. In *International Conference of Information Science and Management Engineering*, pages p567 – 570, 2010.

[9] L. Kroc, A. Sabharwal, and B. Selman. Message-passing and local heuristics as decimation strategies for satisfiability. In *Proceedings of the 2009 ACM symposium on Applied Computing*, pages 1408–1414. ACM, 2009.

[10] M. Mézard, G. Parisi, and R. Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297(5582):812–815, 2002.

[11] Melanie Mitchell. *An Introduction to Genetic Algorithms.* The MIT Press, Cambridge, MA, 1996.

[12] Martin Pelikan. Finding ground states of sherrington-kirkpatrick spin glasses with hierarchical boa and genetic algorithms. In *GECCO 2008: Proceedings of the 10th annual conference on Genetic and Evolutionary Computation Conference*, 2008.

[13] EM Rudnick, JG Holm, DG Saab, and JH Patel. Application of simple genetic algorithms to sequential circuit test generation. *Proceedings of the European Design and Test Conference*, pages 40–45, 1994.

[14] B. Selman, H. Kautz, and B. Cohen. Local search strategies for satisfiability testing. *Cliques, coloring, and satisfiability: Second DIMACS implementation challenge*, 26:521–532, 1993.

[15] G. Syswerda. Uniform crossover in genetic algorithms. In J. D. Schaffer, editor, *Proceeding of the Third International Conference on Genetic Algorithms.* Morgan Kaufmann, 1989.