Choice and Development

Arthur M. Farley
Computer and Information Science Department
University of Oregon
Eugene, OR 97403 USA
art@cs.uoregon.edu

ABSTRACT

The process of development creates a phenotype from one or more genotypes of an individual through interaction with an environment. The opportunity for development to choose a phenotype from a set of alternatives made possible by the individual's genotype(s) has not been widely considered in evolutionary computation. We briefly review recent research on developmental learning, dominance, and hybrid genetic algorithms that has investigated the role of choice in development. A new model of probabilistic development is presented based upon genotypes that encode the probabilities that the various alleles are expressed in the phenotype. The model outperforms a standard, binary haploid model on two families of single-peaked fitness functions in terms of average fitness. The standard model performed better on multi-peaked MAXSAT environments. More research is needed to fully evaluate the new model.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Problem Solving, Search

General Terms: Algorithms

Keywords: genetic algorithms, development, choice, learning, dominance

1. INTRODUCTION

The process of development, whereby a phenotype is created from one or more genotypes of a given individual, has received relatively limited attention in evolutionary computation. A phenotype is that aspect of an individual that is evaluated by an environment. In the vast majority of evolutionary computation applications, a genotype is either evaluated directly or is mapped to a unique phenotype by a deterministic function, this mapping being a reformulation or expansion of a linear genotype to make evaluation easier to perform. This second form of development is often discussed in genetic programming applications, where linear, compressed encodings of programs are expanded into computation trees, cellular automata, or other program forms [17]. In other cases, grammatical or rule -based encodings of complex structures, such as neural networks, have been used, e.g., [27], [9]. These approaches represent a minimal perspective on the development process that only addresses issues of genotype form and can just as well be considered to be part of the evaluation process.

In particular, the possibility for development to select aspects of a phenotype from a set of alternatives made possible by an individual's genotype(s) has not been widely considered. One thread of research where this issue has been addressed is the investigation of the impact of adaptive individuals in evolving populations [5]. Research considering the effects of developmental learning on population dynamics, including investigations into the existence or nonexistence of a facilitative *Baldwin Effect*, has included interaction with an environment to direct choice as part of the development process. Baldwin proposed that learning during development should speed the fixation of adaptive alleles [2]. In recent simulation studies, a local search process during development identifies an improved, *learned* phenotype based upon a given *innate* genotype. To be consistent with the central dogma of biological genetics [21], the *learned* phenotype is used to determine an individual's fitness, but the *innate* genotype is recombined to form the next generation should the individual be selected as parent.

Hinton and Nowlan [13] were among the first to employ simulation as a means for investigating the impacts of learning during development. They introduced ? loci in haploid genotypes, which loci were considered to be learnable. Given a genotype, their learning method tried all combinations of $0, 1$ values for these loci, searching for improvements in fitness of the phenotype. Their research demonstrated a positive impact of developmental learning on average fitness in an extreme environment having only one phenotype with increased fitness, all other phenotypes being equal. Mayley [18] used a binary haploid genotype and developed a phenotype by initially setting it to the genotype and then sequentially trying alternative allele values for all loci, changing the allele that gave the largest increase in fitness; this process continued for some number of cycles, each cycle being termed a *learning operation*. Mayley found a negative, or *hiding*, effect on average innate fitness in an environment having no epistatic interaction among gene loci and a positive impact when such interactions were introduced. Recent research has investigated which conditions give rise to the faciitative Baldwin Effect and which to the hiding effect [29].

GECCO'07 July, 7-11, 2007, London, England, United Kingdom

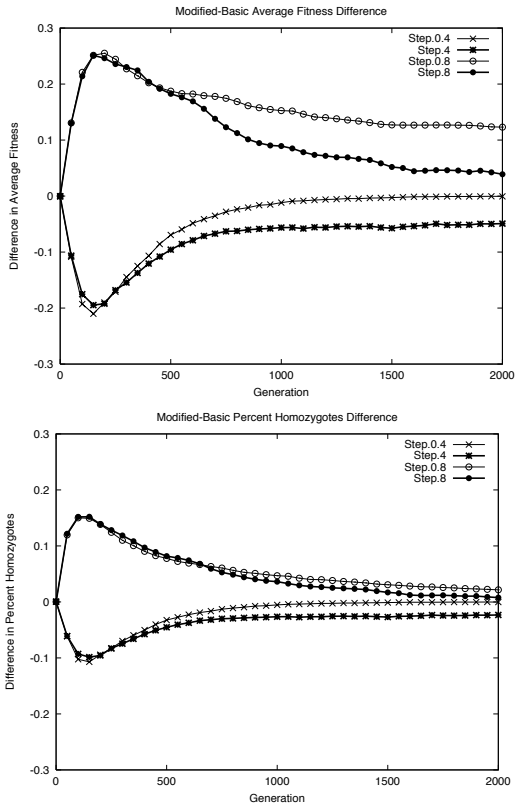

Figure 1: Modified-Basic Differences in Step Average Fitness and Percent Homozygotes for stepSizes 4 and 8.

Diploid representations necessarily give rise to choice during development. At heterozygotes (i.e., loci of the two genotypes where corresponding alleles differ), development must choose which allele to *express* in the resultant phenotype. An *uninformed* choice is one that randomly selects either allele with equal probability, which we will term the *Basic Model* of development. In earlier research [6], we contrast population dynamics generated under the *Basic Model* with that generated by a *Learning Model*. The *Learning Model* sets the phenotype to one of the two genotypes, and then for each heterozygote, considers the other allele for possible expression. If this allele improves fitness, it replaces the original allele in the phenotype. A parameterized family of single-peaked fitness functions, *Step(stepSize)*, were used as environments. Each fitness function compares a given phenotype to a random phenotype called the *point*. With a *stepSize* of $k \geq 1$, k consecutive locations must equal the corresponding *point* locations to *match* a step; there are $genSize/k$ non-overlapping steps. The fitness function returns the percentage of steps that *match*. The *Step* functions represent fitness landscapes having a single peak (i.e., the *point*) with plateaus determined by the steps. Increasing the *stepSize* adds difficulty, as the evolutionary process must accumulate all of a *stepSize* group of alleles before an individual's fitness improves above a current plateau. A *stepSize* corresponds to the degree of *epistatic interaction* between successive gene loci [15].

Figures 1 shows the impacts of developmental learning on the average fitness and genetic diversity of a population of size 1000 given *Step* fitness functions with *stepSize*s of 4 and 8. For purposes of comparing the *Learning Model* to the *Basic Model*, uninformed development is used to generate a phenotype i.e., a *Modified Model*. The plots show the values under the *Modified Model* minus the values under the *Basic Model*, where *stepSize* is indicated by the last legend element; cases with mutations are shown in bold. We observe a phase transition in the impact of developmental learning in these single-peaked environments. With low *stepSize*s, the environmental is *conquerable* by evolution under the *Basic Model*, as the population reaches near optimal average fitness. In these cases, developmental learning has a *Blurring Effect* on evolution, slowing both gain in average fitness and loss of genetic diversity. With higher degrees of epistatic interaction, the environment proves to be *unconquerable* by the *Basic Model*. In these cases, developmental learning exhibits the Baldwin Effect, yielding a faster increase in average fitness and loss of genetic diversity.

From an evolutionary computation perspective, the additional evaluations of the fitness function during developmental learning impact overall genetic algorithm performance and must be considered when comparing overall algorithm efficiency. *Hybrid genetic algorithms* are optimization methods that introduce local optimization methods, some being specific to a given problem domain, that are applied to the individual genotypes determined by evolutionary search [10]. In some cases, local methods are used to fine tune the results determined by the genetic algorithm In others, these methods are applied to all genotypes, with the improved genotypes selected and passed on by the genetic algorithm. These approaches bear close resemblance to the developmental learning research discussed above, though the latter represents a *Lamarckian* perspective on evolution, passing along genotypes that include *acquired* adaptations. There has been a significant range of problems to which hybrid algorithms have been applied. In one experiment investigating automated feature selection for learning algorithms, the hybrid approach was found to outperform simple hill climbing and genetic algorithms [23]. Other applications of hybrid algorithms include the traveling salesman and graph partitioning problems [5], [14].

One area of research in evolutionary computing that has successfully employed hybrid genetic algorithms has been in the creation of artificial neural networks [31]. As such a network is created, both its architecture (i.e., the connections) and weights must be determined. One approach has been to evolve fairly good weights for a given architecture and then use standard neural network training methods, such as backpropagation, to refine the weights and improve ultimate performance [16]. The idea is that evolution efficiently gets the network near a good fitness peak and then standard network training finds that peak. Another approach uses evolution to evolve architectures and then standard network training methods determine the weights [24]. In either case, developmental learning that improves weights through interaction with an environment is a key component of the process.

Choice is seen to be an active component of biological development in a variety of circumstances. These occurrences are said to represent the presence of *phenotypic plasticity* [1]. Research in biology indicates that a significant amount of an individual's genotype encodes information for control of phenotype development based upon available gene alleles. The

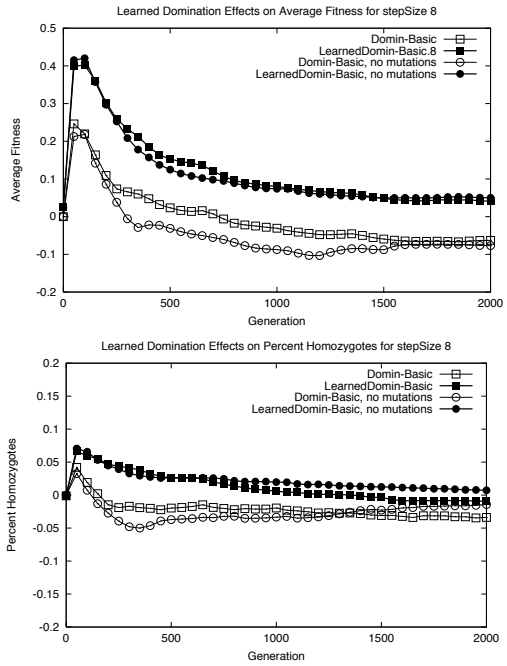

Figure 2: Dominance Differences in Average Fitness and Genetic Diversity

effect of this developmental information can be seen in tissue and organ differentiation throughout a body. Changes in this developmental information can lead to changes in phenotype characteristics even without changes in gene makeup. Of interest is that certain natural systems seem to act to limit this choice by fixing a preference for one allele over another. *Canalization* is the notion that developmental reactions can be adjusted so that they bring about a nearly same outcome regardless of small variations in environmental conditions [28] This concept is supported by the observed constancy of the *wild type* across similar environments [25].

A related evolutionary phenomenon is *dominance*, which has been an element of the theory of evolution since first being identified by Mendel in his classic study of pea populations over 140 years ago [22]. What Mendel noticed was that almost all developed phenotypes for a pure hybrid trait exhibited a single trait value, yet the hidden trait value reappeared in about one-fourth of offspring from those hybrids. Mendel proposed this was due to the non-uniform expression of gene alleles occurring at heterozygous genotype loci, reflecting dominance of the preferred allele over other the possible allele(s). We define *dominance* to be the preferential expression of certain gene alleles at heterozygous diploid loci during phenotype development. The preferred alleles are referred to as *dominant* alleles and the others as *recessive* alleles. Dominance can be understood as a form of developmental canalization, where despite the occurrence of a recessive gene at a heterozygote, the dominant gene is (almost) always expressed [25]. From a biological perspective, dominance can been understood as a narrowing of the reaction range of development [26] at particular gene locations, resulting in stable development patterns.

Dominance that is well-aligned with the environment, i.e., that prefers alleles maximizing fitness, exhibits an *Adaptive Effect*, improving average fitness and maintaining greater genetic diversity over that provided under the *Basic Model* [7]. Two models for the acquisition of well-aligned dominance in diploid individuals were considered, one based solely on homozygote occurrence in ancestors and one involving developmental learning. This form of developmental learning tries alternative alleles for heterozygotes only when there is no dominance preference for the gene locus. By this model, developmental learning does not affect dominance preferences directly. In conquerable environments, this form of developmental learning had no appreciable effect on dominance acquisition or average fitness as the method based on homozygote occurrence already realized well-aligned dominance and near optimal fitness levels. In unconquerable environments, however, developmental learning resulted in more complete acquisition of a well-aligned dominance pattern, which in turn had a Baldwin Effect on population dynamics, improving average fitness with greater loss of genetic diversity. Figure 2 shows results for a *Step* environment with $stepSize$ of 8. We see that the *Domin* model, which acquires dominance solely on the basis of prior homozygotes, results in average fitness that is less than the *Basic Model* in the long term, though facilitating improvement initially. Under the *LearnedDomin* model, improvements in average fitness are greater initially, these increases are persistent through the long-term, asymptotic phase of evolution.

The above results demonstrate that the availability of developmental choice can have positive effects on the rate of improvement in average fitness. In the research reported here, we investigate a probabilistic haploid model that is based upon developmental choice wherein genes reflect the distribution of ancestral development choices.

2. A PROBABILISTIC HAPLOID MODEL

We consider a generational genetic algorithm acting upon haploid individuals. We represent haploid individuals in terms of a single *genotype*, being a sequence of *genes*, each gene having a value chosen from a set of possible *alleles*. Each gene of a *binary genotype* has two possible alleles, which we represent as the set $\{0, 1\}$. A *binary haploid individual* holds a single binary genotype inherited from a single parent or created as a recombination of genotypes from two parents of the previous generation. We also consider *binary probability individuals* that are haploid individuals, each holding a *binary probability genotype*. In such a genotype, each gene locus has as its value a probability p_0 (between 0.0 and 1.0) that the gene will be expressed as a 0 by development. The probability that the gene will be expressed as a 1 is implicit, i.e., $1 - p_0$.

We simulate the processes of evolution by a generational genetic algorithm, defined as follows: [8] [19]:

```
Generational Evolution(pSize, gSize, f, m)
     {p = GenerateInitialPopulation(pSize, gSize);
      Develop(p);
      Evaluate(p, f);
      until(done())
           {p = GenerateNextPopulation(p, m);
            Develop(p);
            Evaluate(p, f);
           }
     }
```

A current population p consists of a set of individuals. We use a fixed-size population model; each successive generation of p will have the same number $pSize$ of individuals. The process *GenerateInitialPopulation(pSize, genSize)* creates a random set of individuals of size $pSize$. In a random *binary haploid population*, each individual is given a binary genotype with equal probability of a 0 or 1 for each gene locaus. In a random *binary probability population*, each individual holds a single binary probability genotype with each gene locus indicating a 50% chance of being developed as a 0. The function *done*() returns true if a desired halting condition is met, e.g., some number of generations have been generated or some number of successive generations of the population do not differ significantly. Each of the capitalized processes can be implemented in a number of different ways, which differences serve as the bases for possible experiments.

The process *Develop(p)* acts on each individual in population p, creating a phenotype from its genotype. In our model, a *binary phenotype* is the same size as the genotype and represents which of the two possible alleles is expressed for each gene locus in the developed individual. Under *HaploidDevelopment*, each locus of the binary genotype of a haploid individual is merely copied to its phenotype, i.e., an identity mapping. Under *ProbabilisticDevelopment*, each gene locus is developed as a 0 according to the probability found in the binary probability genotype or is developed as a 1, otherwise. *ProbabilisticDevelopment* impacts an individual's genotype, modifying the underlying genetic content of the individual. If a 0 is expressed at a given gene locus, p_0 is increased; if a 1 is expressed, it is decreased. How much of an increase or decrease is left as an open degree of freedom under this general model. We compare three possibilities in an experiment reported below.

An environment is represented as a *binary fitness function*, which is an arbitrary function mapping from a binary phenotype to a real value in the range 0.0 to 1.0. The value of a fitness function when applied to an individual's phenotype indicates the individual's degree of adaptation to the environment, i.e., the higher the value the better adapted is the individual. The process *Evaluate(p, f)* applies a binary fitness function f to the phenotype of each individual in population p and associates the resulting fitness value with the individual.

The process *GenerateNextPopulation(p, m)* creates a new generation of population p, as follows:

```
GenerateNextPopulation(p, m)
      {parents = SelectParents(p);
       p = Recombine(parents);
       return Mutate(p, m);
      }
```

The process *SelectParents(p)* is that aspect of our model where notions of natural selection and selective pressure are captured. It selects pairs of parents whose offspring will form the next generation. We use *FitPropSelection* whereby an individual i is selected to be a parent with probability f_i/F_p, where f_i is the fitness value of individual i and F_p is the sum of fitness values of all individuals in population p. This has been called *fitness proportionate selection* in the genetic algorithm literature [19]. The process *Recombine(parents)*, considers each pair of parents and creates a new individual having two genotypes, one inherited from each parent as a result of recombining each parent's two genotypes. We use *UniformRecombination(parents)* where one of the two values from the genotypes of the two parents is selected with equal probability at each location. This recombination method has been called *uniform crossover* [8]. Finally, the genotype of each individual in the new population may be altered by mutation, implemented as *Mutate(p)*. Each binary gene of the genotype of each individual in new population p is changed to the other allele according to a given, per locus mutation probability m. In the case of a binary probability genotype, probability p_0 is changed to $1 - p_0$.

3. EXPERIMENTAL FRAMEWORK

In addition to the *Step* family of fitness functions defined above, a related $NKStep$ family of fitness functions is considered. Rather than the non-overlapping steps of a *Step* fitness function, an $NKStep$ function has every bit being the first bit of a contiguous $stepSize$ step. Given a phenotype of n bits, there are thus n steps that are matched to the *point* of the fitness function. Again the percentage of steps that completely match the *point* is the fitness value. As noted, these environments present single-peaked fitness landscapes of varying degrees of difficulty. To consider more complex settings, we also investigate $MAXSAT$ fitness where a binary phenotype represents boolean assignments to variables and fitness of an assignment is the percentage of clauses satisfied in randomly generated 3-Sat problems having 100 variables and 300 and 500 clauses, giving problems on different sides of the satisfiability threshold [11].

Two measures of a population are considered at each generation. The first corresponds to a population's level of adaptation to an environment as measured by its *average fitness*, being the average of all individuals' fitness function values. Expect average fitness is $1/2^k$ for a $stepSize$ of k in a random, initial population for both of our step related fitness functions. The other measure addresses the genetic diversity of a population. The diversity measure is *percent polymorphic*, defined as the percentage of gene loci that remain polymorphic in a haploid population. A location is considered polymorphic if less than 99% of its alleles are equal [21]. Percent polymorphic is 1.0 in a random, initial population. In the case of our *BinaryProbability* population, we consider a particular location of a given individual to be *fixed* if $p_0 < 0.01$ or $p_0 > 0.99$. A location is polymorphic if less that 99% of its alleles are fixed and equal.

There are several parameters that will not change throughout the experiments reported here. We have set the population size to be 1000 for all experiments. This is a relatively small, finite population that makes the simulation experiments feasible, but that is hopefully large enough so that results are not dominated by small population effects, such as genetic drift [12]. Another factor that is not varied is genotype size, i.e., number of loci or genes in a genotype. The genotype size is set to 100, again a compromise between computational efficiency and genetic diversity. The mutation rate is set so that an average of 10 alleles over the whole population for each generation are modified by mutation, yielding a per locus mutation probability of 10^{-4}, given the population and genotype sizes we have chosen. This is in the range of observed mutation rates in nature [21]. The experiments are run for 2000 generations; each experiment is repeated 20 times to generate average values and standard deviations. Values for the two population measures are gathered every 50 generations.

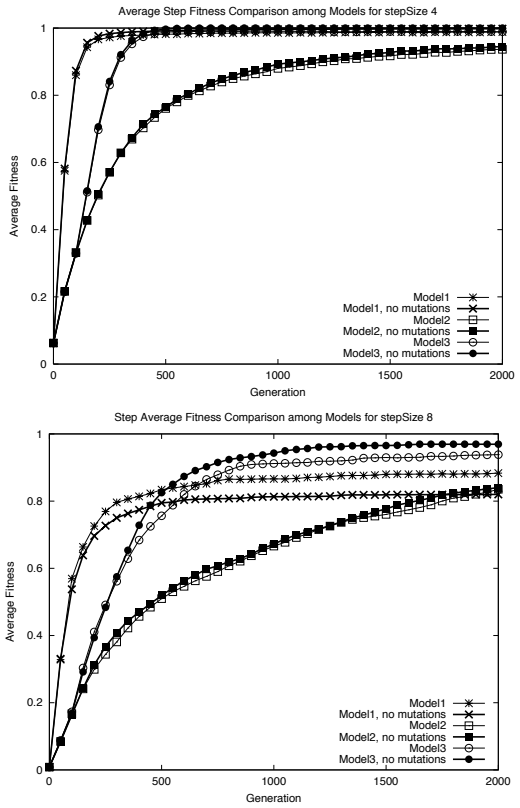

Figure 3: Comparisons of Average Fitness for Three Probabilistic Models on Step Fitness with stepSizes 4 and 8

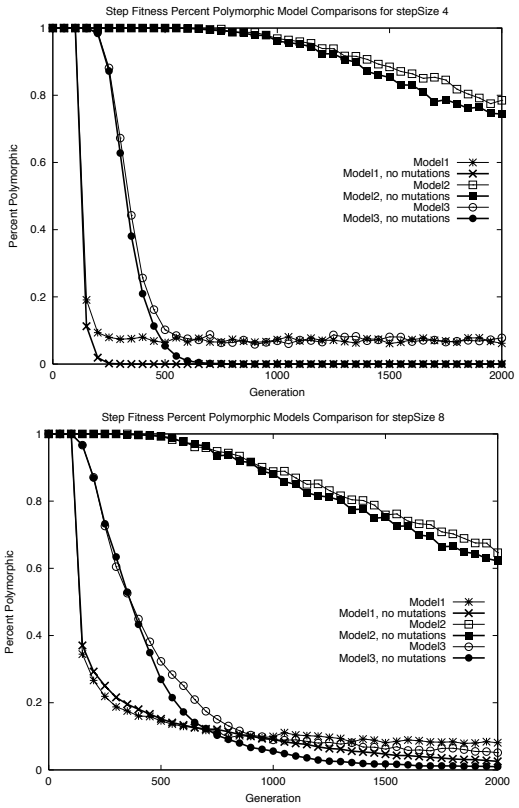

Figure 4: Comparisons of Percent Polymorphic for Three Probabilistic Models on Step Fitness with stepSizes 4 and 8

4. RESULTS

As noted above, how to update p_0 based upon a current development choice is a key aspect of the probabilistic model. If p_0 were not changed, it would remain at 0.50, and no evolution of the initial, random population would occur. In our first experiment, we consider three related approaches to p_0 update. All three approaches update p_0 through a common sequence of values. When a consecutive series of 0's is expressed, p_0 increases through the sequence of values $(k-1)/k$, i.e., 0.50, 0.67, 0.75, 0.80, 0.833, etc., with k increasing each time that a 0 is expressed. Similarly, if a series of 1's is expressed, the probability p_0 decreases through the sequence of values $1/k$, i.e., 0.50, 0.33, 0.25, 0.20, 0.166, etc., with k increasing each time that a 1 is expressed. The three approaches differ as to the update made when the currently less probable allele is expressed. By *Model1*, if the less probable allele is chosen, we simply reduce k by one, e.g., 0.80 becomes 0.75 or 0.20 becomes 0.25. By *Model2*, the probabilities for the two alleles are equalized, returning to the starting, unbiased value, i.e., p_0 becomes 0.50, thereby giving significantly more weight to a recent development choice. *Model3* uses the update scheme of *Model2* but in addition allows for *fixation* of a gene locus. If p_0 reaches100/101, it is set to 1.0, thus becoming a $fixed$ location. Similarly, if a series of 1's is expressed and p_0 reaches 1/101, p_0 is set to 0.0, becoming a $fixed$ location. Results presented in Figure 3 indicate the contrasting aspects of performance that these models produce. $Model1$ quickly leads to alleles becoming all but fixed at a particular value. If a less likely allele is chosen by development, the probability still remains high that the more likely allele will be selected in the future, given the individual is selected to mate. This development pattern leads to quick convergence at near optimal fitness values for smaller $stepSize$s, such as 4 in the upper plot, which environments are relatively easy for evolution to conquer. When $stepSize$ is increased to 8, however, the relative inability to recover from development choices that are not optimal leads to convergence at lower average fitness values. Looking at the other two models, we see that $Model2$, which resets allele probabilities to 0.50 when the less likely allele is chosen, has trouble converging to near optimal fitness, producing slower improvement in average fitness. $Model3$, as a compromise, demonstrates that an ability to reset probabilities based upon development during early generations, accompanied by an ability to fix an allele when its probability becomes high enough, is able to realize near optimal average fitness in environments with lower $stepSize$, albeit at a slower rate than $Model1$, and is able to outperform the other two models as $stepSize$ increases, as indicated for the results with a $stepSize$ of 8. Figure 4 presents the corresponding percent polymorphic results which support this analysis, as

Figure 5 compares the performance on average fitness between $Model3$ and the standard binary $Haploid$ model for $Step$ and $NKStep$ environments with $stepSize$s of 4, 8, and 10. We see $Model3$, while trailing the standard haploid

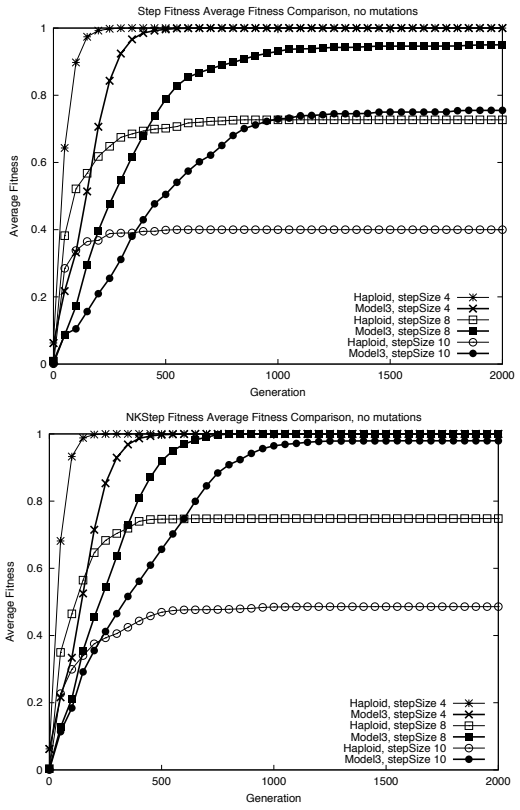

Figure 5: Comparisons of Average Fitness for Haploid and Probabilistic Models on Step and NKStep Fitness Functions with stepSizes 4, 8, and 10

model for smaller *stepSize*s such as 4, significantly outperforms the standard haploid model for higher *stepSize*s. The environments become unconquerable for the standard model, while $Model3$ is able to continue to realize near optimal average fitness for the $NKStep$ environments. The performance of the standard haploid model is closely related to that of the probabilistic $Model1$, which quickly fixes alelle values within individuals.

Considering more complex, multi-peaked environment represented by the $MAXSAT$ fitness function, we find a different picture. Figure6 shows that the standard haploid model consistently outperforms the probabilistic $Model3$ in terms of average fitness; in the long term, the differences become minimal.

5. CONCLUSION

We have presented a new haploid model for genetic algorithms that involves the probabilistic development of individuals' phenotypes based upon genotypes that encode the probabilities of allele expression and reflect the impacts of the recent history of ancestral development. Our model bears certain resemblance to ant colony optimization algorithms and related methods of swarm intelligence [4]. Both notions are based upon a probabilistic knowledge representation that guides the generation of certain solution elements and is altered by choices made during a generation. However, as the name implies, ant colonies and swarms combine their knowledge into one probability matrix according to

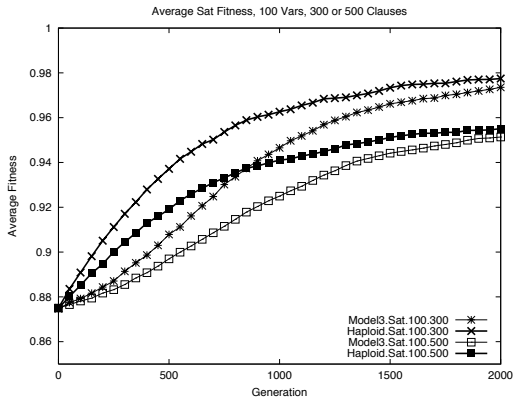

Figure 6: Comparison of Average Fitness for Haploid and Probabilistic Models on MAXSAT Fitness with 300 and 500 clauses.

which a next individual or next generation of individuals is created. Under our model, individuals carry their own history of recent, ancestral gene expression represented as probabilities that are used for phenotype development. The developed phenotypes are evaluated and then serve as basis for the parental selection that drives evolution. The individual probabilistic genotypes undergo crossover and mutation by standard, haploid genetic operations during mating and creation of the succeeding generation.

We present examples of our model applied to single-peaked *Step* and $NKStep$ environments with varying degrees of epistatic interaction and to multi-peaked MAXSAT environments. As *stepSize* increases in the *Step* and $NKStep$ environments, the new probabilistic model outperforms the standard binary haploid model in terms of average fitness achieved. On the MAXSAT environments, the results are reversed, with the probabilistic model approximating the standard haploid model only after a significant number of generations. More experience is needed with the new model to better understand its behavior, to consider other methods for genotype probability update, and to gauge its general applicability to problems of interest. Efficient encodings and methods allowing the model to represent non-binary genotypes are also needed.

6. REFERENCES

[1] Ancel, L.W., "Undermining the Baldwin expediting effect: Does phenotypic plasticity accelerate evolution?", *Theoretical Population Biology*, 58, 307-319, 2000.

[2] Baldwin, J.M., "A new factor in evolution", *American Naturalist*, 30, 441-451, 1896.

[3] Belew, R.K. and Mitchell, M., *Adaptive Individuals in Evolving Populations*, Santa Fe Institute Proceedings, Addison Wesley, Reading, MA, 1996.

[4] Bonabeau, E., Dorigo, M. and Theraulaz, G., *Swarm Intelligence: From Natural to Artificial Intelligence*, Oxford Univ. Press: Oxford, 1999.

[5] Bui, T.N. and Moon, B.R., "Genetic algorithms and graph partitioning", *IEEE Trans. Computers*, 45(7), (1996), 841-855.

[6] Farley, A.M., "Developmental learning in single-peaked environments", in submission, 2007.

[7] Farley, A.M., "Domination in development", in submission, 2007.

[8] Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, Cambridge,MA. (1989).

[9] Gruau, F., Whitley, D., and Pyeatt, L., "A comparison between cellular encoding and direct encoding for genetic neural networks", *Proc. Genetic Programming Conf.*, (1996), 81-89.

[10] Hacker, K. and Lewis, K., "Robust design through use of a hybrid genetic algorithm", *Proceedings of DETC'02* ASME, 2002, 1-10.

[11] Hajiaghayi, M. T., Coppersmith, D.,Gamarnik, D. and Sorkin, G. B., "Random MAXSAT, random MAXCUT, and their phase transitions", *Proceedings of Fourteenth ACM-SIAM Symposium on Discrete Algorithms (SODA)*, (2003), 364-373.

[12] Hedrick P.W., *Genetics of Populations*, Jones and Bartlett, Sudbury,MA, (2000).

[13] Hinton, G.E and Nowlan, S.J., "How learning can guide evolution", *Complex Systems*, 1, p495-502, 1987.

[14] Jog, P., Suh, J., and Gucht, D., "The effect of population size, heuristic crossover, and local improvement on a genetic algorithm for the traveling salesman problem", *Proceedings Int. Conf. on Genetic Algorithms*, (1989) 110-115.

[15] Kauffman, S.A., *The Origins of Order*, New York: Oxford University Press, (1993).

[16] Kinnebrock, W., "Accelerating the standard backpropagation method using a genetic approach", *Neurocomputing*, 6, (1994), 583-588.

[17] Koza, J. et. al., *Genetic Programming IV: Routine Human-Competive Machine Intelligence*, Kluwer Academic Publishers, (2003)

[18] Mayley, G. "Guiding or hiding: Explorations into the effects of learning on the rate of evolution", In *Proceedings of the Fourth European Conference on Artificial Life 97*, 135-144. 1997

[19] Mitchell, M., *An Introduction to Genetic Algorithms*, MIT Press, Cambridge, Ma., (1996).

[20] Maynard Smith, J., *The Theory of Evolution, 3rd Edition*, New York : Cambridge Univ. Press, (1993).

[21] Maynard Smith, J., *Evolutionary Genetics, 2nd Edition*, New York : Oxford University Press, (1998).

[22] Mendel, G., "Experiments in plant hybridization", 1866, reprinted in Stern, C. and Sherwood, E.R. (eds), *The Origins of Genetics. A Mendel Source Book*. San Francisco: Freeman, 1966, see also http://www.mendelweb.org/Mendel.html.

[23] Oh, I-S., Lee, J-S. and Moon, B-R., "Hybrid genetic algorithms for feature selection", *IEEE Transactions on Pattern Analysis and Machine Intelligence* , 26(11), (2004), 1424-1437.

[24] Sarkar, M. and Yegnanarayana,B ., "Feedforward neural networks configuration using evolutionary programming", *Proceeding IEEE Intl. Conf. on Neural Networks*, (1997), 438-443.

[25] Siegal, M.L., and Bergman, A. "Waddington's canalization revisited: Developmental stability and evolution", *Proc. Nat. Acad. Sci*,99(16), (2002) 10528-10532.

[26] Simpson, G.G.,"The Baldwin Effect", *Evolution*, 7, 1953, 110-117, reprinted in [5].

[27] Vonk, E., Jain, L., and Johnson, R., "Using genetic algorithms with grammar encoding to generate neural networks", *Proceeding IEEE Intl. Conf. on Neural Networks*, (1995), 1928-1931.

[28] Waddington, C.H., "Canalization of development and the inheritance of acquired characters", *Nature*, 3811, 1942, 563-565, reprinted in [5].

[29] Weber, B.H. and Depew, D.J., *Evolution and learning: The Baldwin effect reconsidered*, Cambridge, MA : MIT Press, 2003.

[30] Wright, S. "Evolution in Mendelian populations", *Genetics*, 16, (1931), 1423-1447, (1999).

[31] Yao, X.. "Evolving artificial neural networks", *Proceedings of the IEEE*, 87(9), (1931), 97-159.