

AN ANALYSIS OF GENETIC-BASED PATTERN TRACKING AND COGNITIVE-BASED COMPONENT TRACKING MODELS OF ADAPTATION

Elaine Pettit and Dr. Kathleen M. Swigger

North Texas State University

ABSTRACT

The objective of this study was a comparison of the effectiveness in adapting to an environment of populations of structures undergoing modification by four different models: 1) Holland's (2) genetic operator model; 2) a cognitive (statistical predictive) model; 3) a random point mutation model; and 4) a control (non-altered) model.

INTRODUCTION

Holland (3) has reviewed the prolonged success of lifeforms in adapting to an environment through evolution. The biological organism is faced with testing a large set of possible genetic expressions in its offspring by means of environmental interaction with a relatively small subset of realized structures (its own genotype). Nonlinearity and epistatic interactions among gene sets complicate the problem of achieving a successful, if not optimal, genetic complement in offspring. Holland has mathematically hypothesized that genetic operators (e.g., crossing-over) exploit the optimization of reproductive fitness (number of offspring) by a means he terms intrinsic parallelism. Intrinsic parallelism is the testing of a large pool of schemata (the set of all partitions and combinations thereof of a prototypical structure, or genome) by means of a much smaller subset of realized structures. More simply, consider the structure A consisting of a string of six binary digits, (1 0 1 1 0 1). Each binary digit may be considered to be a detector in an off/on state (i.e., comparable to alleles in genetics). Structure A is a member of a set of structures α which includes all possible strings of six binary digits. There exists a superset E which is the set of strings of length six composed of concatenations of {1,0,#}, where # represents a "don't care" position, i.e., its value as 0 or 1 is irrelevant. For example, let $E \in E$ be (1 0 # 0 1). This E is termed a "schema," and all possible schemata compose \bar{E} , the "pool of schemata." The structure A (1 0 1 1 0 1) is an instance of the schemata (1 0 # # 0 1) and (# # # # 0 1), but not of (0 # # 0 # 1). Now consider a structure Environ (0 0 1 0 0 1) which represents the "state" of an environment. Fitness, or performance, may then be defined as the number of matching elements between structure A and structure Environ with a one-to-one correspondence: in this illustration, the fitness of A would be 4. Schemata represent the contribution to fitness of single detectors (i.e., alleles) as well as of combinations of

detectors. A subset of structures from α constitutes a population. It is, by definition, the goal of adaptation to modify these structures in order to optimize the fitness of the population. Holland has shown that the genetic operators of crossing-over, inversion, and, to a limited extent, mutation are highly successful in 1) testing a large number of possible schemata through modifications on a much smaller number of realized structures, and 2) exploiting local optima on the way to achieving the global optimum without becoming entrapped, as opposed to what occurs in the simple hill-climbing technique of heuristic search. The purpose of this study was to garner empirical evidence from an abstract computer implementation of Holland's model with regard to alternative models. LISP was the language of choice due to the power of its list-processing functions.

ELEMENTS OF MODEL CONSTRUCTION

I. Primary Data Structures

Environment (ENVIRON) - a list of 12 randomly-selected binary digits (example - (1 1 0 0 0 0 1 1 1 0 1 1)).

Matrices (MATRICES) - a list of transition matrices for each binary digit in the environment. It is used to simulate a Markov-chain type stochastic variation in the states of the environment. (Example - ((.5 .3) (.2 .9) (.6 .4) . . . (.3 .7)). (.5 .3) would represent the transition matrix

	0	1
0	.5	.5
1	.3	.7

Populations - For each model, a list of 12 sublists of 12 randomly selected binary digits, each of these sublists representing a structure. (Example - ((1 1 0 0 1 1 1 1 0 0 0 1) (0 0 1 0 0 1 1 1 0 0 1 0) . . . (0 1 1 1 1 1 0 0 0 1 0 0))).

II. Primary Measurements

- 1) Adaptation - modification of structures to improve performance
- 2) Performance (fitness) - number of one-to-one matches between a structure in a population and the environment, for a given state of the environment.
- 3) Averaged Population Performance (population fitness) - the average number of matches

over the population for a given transition of the environment.

- 4) Tracking - the change in averaged population performance over a stated number of environment transitions.

III. Description of Algorithms for Models

- 1) BETA - the population undergoing the genetic operations of crossing-over, inversion, and mutation (adapted from (2)).

A. General Model Algorithm: Initialize population, environment, and transition matrices. Find the performance for each structure in BETA. Call it MU(i). Define the random variable RAND on $\{1, \dots, M\}$ by assigning probability $MU(i) / \sum_{i=1}^M MU(i)$ to each

structure in BETA, where M is the number of structures in the population. Make M trials of RAND, each time storing the structure at position RAND in BETA at successive positions in auxiliary list BPRIME. For each structure in BPRIME apply inversion and mutation, and, if it is a structure in an even-numbered position, cross it with the immediately preceding structure. Set BETA to BPRIME. Apply transition matrices to environment. Repeat all steps except initialization for desired number of transitions.

- B. Algorithm for Crossing-Over: Make a trial of RAND on $\{0, \dots, \ell\}$, where ℓ is the number of positions in each structure. If RAND = 0 or RAND = ℓ then no crossing-over takes place. Otherwise, take positions 1 through RAND of hth structure and append positions RAND + 1 through ℓ of (h-1)th structure. Likewise, take positions 1 through RAND OF (h-1)th structure and append positions RAND + 1 through ℓ of hth structure.

(Illustration of mechanism):

hth Structure:	1	0	0	0	0	0	0	1	
(h-1)th structure:	1	1	1	1	0	1	0	1	

After crossing over:

hth structure:	1	0	1	1	0	1	0	1	
(h-1)th structure:	1	1	0	0	0	0	0	1	

- C. Algorithm for Inversion: Make 2 trials of RAND on $\{1, \dots, \ell\}$ and designate the outcomes X1 and X2, respectively. Take the segment of the structure from position MIN (X1, X2) through position MAX (X1, X2), reverse it, and reinsert it into the structure.

(Illustration of mechanism):

			minX			maxX			
Structure:	1	0	1	0	1	0	1	0	
After inversion:									
Structure:	1	0	0	1	0	1	1	0	

- D. Algorithm for Mutation: Make a trial of RAND on $\{1, \dots, \ell\}$ and designate the outcome X. Make another trial of RAND on integers 0,1 and designate the outcome CHANGE. Take position X of the structure and change it to CHANGE.

(Illustration of mechanism):

Structure: 0 0 0 0 1 0 0 X = 3; Change=1
After Mutation: 0 0 1 0 1 0 0

- 2) MEMPOP - the "cognitive" model. This routine kept track of the number of times in each position that the structure failed to match the environment for each transition. The routine then calculated the probability from the above frequency that it should "flip" that position, and adjusted each position accordingly.

General Algorithm: Initialize population, environment, and transition matrices. Create MEMORY, a list of 12 elements, one for each position of each structure in MEMPOP, and set to 0. For each structure in MEMPOP, do the following for the desired number of transitions: 1) find performance with current environment; 2) for each position that does not match, add 1 to the sum at the corresponding position in MEMORY; 3) for each position in the structure, make a trial of RAND distributed $U(0,1)$, and if RAND is less than the sum at the corresponding position in MEMORY divided by the number of transitions, then flip that bit. Find the averaged population performance over all transitions.

- 3) RANDPOP - underwent random point mutation by algorithm for mutation above, with number of mutations being generated from POISSON(1). Structures were taken through transitions separately, as in the algorithm for MEMPOP above.
- 4) CONTROLPOP - underwent no modification. Structures were taken through transitions separately, as in the algorithm for MEMPOP above.

TESTING

Part I. Comparisons Among All Four Models

All populations were initialized to the same set of structures and encountered the same initial environment and the same transition matrices. The entire set of BETA (genetic model) was necessarily carried through the transitions at the same time, since structure interaction is inherent in the genetic model. Each structure of the other populations was carried through the transitions singly.

The averaged performance of the population was calculated for each transition. Two runs, each of which consisted of twenty-five transitions, were made with the same initial populations and environment but with different transition matrices.

Statistical analysis of possible differences in the 25 averaged population performances among the groups was performed under the following assumptions: 1) that the underlying distribution, though most probably binomial, could be approximated by a normal distribution for $N=25$ (4); 2) that the data values, although obviously correlated for the BETA population, could be treated as essentially independent due to the large number of mutually exclusive stochastic events determining the outcome of the averages. All statistical analyses were run on SAS. A parametric ANOVA was performed to test for differences in the mean of the averaged population performance among the 4 groups. Means were then grouped into equivalence classes by Duncan's multiple range test. To test for the validity of the conclusion without the above two assumptions, a Kruskal-Wallis nonparametric analysis of variance by rank test was also run. Alpha was .05 for all tests.

Table 1 - Summary Data on Comparison of Models

Test 1: Values in Transition Matrix Set to Simulate Environment with Frequent Perturbations

Results: $\alpha = .05$; H_0 : The means of the averaged population performance of all groups are not significantly different. (H_0 for nonparametric test: The rank scores are not significantly different.)

Run 1:

Matrices = ((.5.2) (.9.1) (.4.8) (.2.3) (.7.6)
(.5.5) (.8.5) (.1.2) (0.0 .3) (.4.7)
(.8.4) (.5.9))

ANOVA: AMONG MS WITHIN MS
2.08965 0.344845
F VALUE PROB>F
6.06 0.0009

Reject H_0 .

Duncan's MRT: (Boxes indicate equivalence classes.)

Group	MEMPOP	BETA	CONTROLPOP	RANDPOP
Mean	6.3672	5.9500	5.7892	5.7240

Kruskal-Wallis (Chisqr approximation):

GROUP	SUM OF SCORES
BETA	1254.00
MEMPOP	1751.00
CONTROLPOP	1064.50
RANDPOP	980.50
CHISQR=16.99	PROB>Chisqr = .0007

Reject H_0 .

Run 2:

Matrices = ((.4.6) (.3.9) (.5.8) (.7.5) (.2.1)
(.3.3) (.4.6) (.5.1) (.4.7) (.4.2)
(.9.9) (.3.5))

ANOVA: AMONG MS WITHIN MS
2.62818 0.278441
F VALUE PROB>F
9.44 .0001

Reject H_0 .

Duncan's MRT:

Group	MEMPOP	CONTROLPOP	RANDPOP	BETA
Mean	6.7064	6.2632	6.0932	5.9628

Kruskal-Wallis (Chisqr approximation):

GROUP	SUM OF SCORES
BETA	979.00
MEMPOP	1255.50
CONTROLPOP	1805.50
RANDPOP	1010.00
Chisqr = 20.86	PROB>Chisqr = .0001

Reject H_0 .

Test 2: Values in Transition Matrix Set to Simulate Environment in Steady State With Rare Perturbations

The same procedure as in Test 1 was used with the same initial populations but with a different initial environment and different transition matrices.

Matrices = ((.1.1) (.1.1) (.9.9) (.1.1) (.9.9)
(.1.1) (.1.2) (.1.1) (.1.1) (.9.1)
(.1.9))

Results: $\alpha = .05$; H_0 : same as Test 1

ANOVA: Among MS=1.09459 Within MS =0.23934
F Value = 4.57 Prob>F=0.0050
Reject H_0 .

Duncan's MRT:

Group	MEMPOP	BETA	CONTROLPOP	RANDPOP
Mean	6.5684	6.2028	6.1328	6.1304

Kruskal-Wallis (Chisqr approximation):

GROUP	SUM OF SCORES
BETA	1197.00
MEMPOP	1743.50
CONTROLPOP	1040.00
RANDPOP	1069.50
CHISQR = 15.32	PROB>Chisqr = .0016

Reject H_0 .

Test 3: Transition Matrices Set to Simulate Environment in Absorbing State (Steady State With No Fluctuation)

The same procedure as in Test 2 was followed but with transition matrices as follows:

Matrices = ((00) (00) (00) (00) . . . (00))

Results: $\alpha = .05$; H_0 : Same as Test 1
 ANOVA: Among MS=41.4344 Within MS=0.366581
 F Value = 113.03 Prob>F=0.0001
 Reject H_0 .

Duncan's MRT:

Group	MEMPOP	BETA	CONTROLPOP	RANDPOP
Mean	8.0604	7.9040	6.1632	5.4596

Kruskal-Wallis (Chisqr approximation):

GROUP	SUM OF SCORES
BETA	1843.50
MEMPOP	1868.50
CONTROLPOP	997.50
RANDPOP	340.50
CHISQR=77.23	Prob>Chisqr = .0001

Reject H_0 .

DISCUSSION

With regard to obtaining and processing information from an environment with constant, albeit rare, perturbation, the genetic model performed significantly less well than the cognitive model. The mean of its averaged population performance was no better than that of a control population with no tracking mechanism. Rank scores from the Kruskal-Wallis test reflected a similar relationship. Likewise, random point mutation did not produce results significantly different from the genetic or control models. In a stochastically fluctuating environment, the cognitive model tracked significantly better than all the others, but at a very high price in computational overhead (see following discussion).

In the third test, that of an environment with no fluctuation, the genetic model performed significantly better than the control and random models, and as well as the cognitive model. Hence, the genetic model did not appear to track an environment very well on a short-term basis, but in matching a highly stable environment, it performed, on the basis of structural information, as well as the model possessing the highest level of information concerning each individual bit. As can be noted from the software algorithms, computational overhead for the cognitive model may be assessed as manipulating n structures times x bits per structure. For most applications, x is greater than n , giving a rough complexity estimate of $O(n^2)$. This complexity is reflective not only of arithmetic operations, but of calls to the random number generator (also roughly n^2). The genetic model, however, retains and manipulates information on the basis of the structures themselves, using the "reproductive" fitness as the selection criterion for proportion of inclusion in the next generation. Computational complexity is thus roughly $O(n)$, and the number of calls to the random number generator is also $O(n)$. As Holland mathematically deduces, the genetic operators manipulate the structures so as to 1) increase the averaged population performance, and 2) to test large instances of schemata. The first consequence is structure-based information storing and updating. The second is bit-based, without demanding individual bit updating, or even a query

concerning the status of individual bits in an applications-oriented measure of performance (one would simply measure the "performance" in terms of a desired property of the system). Therefore, the results of these tests provide empirical evidence in support of Holland's proposals.

Part II. Comparisons Between and Within the Genetic and Cognitive Models of Changes in Fitness over Transitions (No Environmental Fluctuation)

The purpose of this section of the experiment was to see if and how the genetic and cognitive models produce an increase in fitness over time. For all the following tests, BETA and MEMPOP were set to the same initial population. Transition matrices were set to zero. Two runs of fifty transitions each were made with two different initial environments. The averaged population performances were calculated for each transition as in Part I. The same statistical assumptions as in Part I were made concerning the independence, normality, and, initially, the homoscedasity of the underlying sample distribution.

Table 2 - Comparison of Mean Averaged Performance Between Genetic and Cognitive Models

Test 4: Comparison of Mean Averaged Population Performance Over 50 Transitions Between Genetic and Cognitive Models

$\alpha = .05$; H_0 : The means of the averaged population performances of the two models are the same (or rank scores are same for the nonparametric test).

Results:

Run 1:

ANOVA: Among MS=10.0109 Within MS=0.410682
 F Value = 24.38 Prob>F=.0001
 Reject H_0 .

Duncan's MRT:

Group	BETA	MEMPOP
Mean	8.9016	8.2688

Wilcoxon 2-Sample Test (Normal Approximation):

GROUP	SUM OF SCORES
BETA	3335.50
MEMPOP	1714.50
$Z = 5.5840$	$PROB> Z = 0.0000$

Reject H_0 .

Run 2:

ANOVA: Among MS=3.6864 Within MS=0.619144
 F Value=5.95 Prob>F=0.0165
 Reject H_0 .

Duncan's MRT:

Group	BETA	MEMPOP
Mean	8.2994	7.9154

Wilcoxon 2-Sample Test (Normal Approximation):

GROUP	SUM OF SCORES
BETA	2936.50
MEMPOP	2113.50
$Z = 2.8334$	$PROB> Z = 0.0046$

Reject H_0 .

Conclusion:

Reject H_0 in both runs and conclude that the averaged population performance of the genetic model is significantly greater than that of the cognitive model.

From the 2 runs of 50 transitions each, the change in performance between successive transitions was calculated by subtracting from each population performance (except the first) the value of the one immediately preceding it (98 observations). $\alpha = .05$; H_0 : The means of the change in population performance between successive transitions are the same for both models (or rank scores of differences are same).

Table 3 - Comparison of Mean Changes for Genetic and Cognitive Models

Test 5: Comparison of Mean Change in Performance Between Successive Transitions for Genetic and Cognitive Models

Run 1:

ANOVA: Among MS=.01805 Within MS=.246721
F = .07 PROB>F=.7874

Do Not Reject H_0 .

Group	Mean Difference
BETA	.071429
MEMPOP	.044286

Wilcoxon 2-Sample Test (Normal approximation):

GROUP	SUM OF SCORES
BETA	2491.00
MEMPOP	2360.00

$Z = .4618$ PROB>|Z| = .6442

Do Not Reject H_0 .

Run 2:

ANOVA: Among MS=.0372255 Within MS=.422695
F = .09 PROB>F = .7673

Do Not Reject H_0 .

Group	Mean Difference
BETA	.06
MEMPOP	.02

Wilcoxon 2-Sample Test (Normal approximation):

GROUP	SUM OF SCORES
BETA	2500.00
MEMPOP	2351.00

$Z = .5258$ PROB>|Z| = .5990

Do Not Reject H_0 .

Conclusion: Do not reject H_0 . Conclude that the mean change in performance between successive transitions is the same for the genetic and cognitive models.

From the above analysis, it was noted that the MS within the groups was greater than that between them. In order to study the variations of the differences in performance within each group, a series of paired-sample t-tests were run on BETA and MEMPOP separately for both sets of the 50-transition data. In computing change over time, "lag" is defined to be the number of transitions between the two environment states for which the population performances are being subtracted. For example, lag 1 is the difference between a value

and its immediate predecessor as above.

Table 4 - Differences in Performance Within Groups

Test 6: Paired Sample t-test for Individual Group Change in Performance (Lag=1)

$\alpha = .05$, H_0 : The mean difference between successive performances is 0 for the group under consideration.

GROUP	RUN 1	RUN 2
BETA	t=1.16 Prob>/t/=.2518	t=.91 Prob>/t/=.3696
MEMPOP	t=.56 Prob>/t/=.5788	t=.18 Prob>/t/=.8585

Conclusion: For all cases, do not reject H_0 . Conclude that the mean change in performance between immediately successive transitions is not significantly different from 0 for both the genetic and cognitive models.

Test 7: Effects of Different Lags on Significance of Change in Performance (Paired Sample t-test)

$\alpha = .05$, H_0 : A lag of X units has no effect on change in performance between transitions (change = 0). (lag steps not given were not significant up to the final one)

RUN 1				RUN 2			
GROUP	LAG#	t	Pr>/t/	LAG#	t	Pr>/t/	
BETA	3	1.57	.1226	3	1.73	.0907	
				5	2.14	.0382	S
	12	1.97	.0567				
	15	2.06	.0467				S
MEMPOP	21	-1.56	.1301	9	-.69	.4955	
	23	-1.24	.2267	20	-2.30	.0291	S
							(in negative direction)

*S means significant

Conclusion: Conclude that a smaller amount of lag time is required for the change in performance to be significant for BETA than for MEMPOP. Also, the t-statistic was in the positive direction for BETA indicating net improvement, whereas it was negative for MEMPOP indicating net deterioration.

SUMMARY AND CONCLUSIONS

This experiment consisted of two major sections: 1) model performance in stochastically-changing environments with varying rates of fluctuation, and 2) model performance in a non-changing environment.

In the first case, it was concluded that the genetic model performed poorly in tracking the changing environments even when the rate of fluctuation was slow. Indeed, it did no better than simple chance (the control model). Likewise, the random point mutation model fared no better, although it has been used to introduce stochastic and, as is hoped, eventually progressive change in the performance of adaptive systems, in fields from traditional biological evolutionary theory to checkers-playing programs. The cognitive model, as was expected, performed significantly better in tracking the environment, but at an unrealistic computational cost. It would appear that none of these models offers any gain over established AI routines in tracking independent components of a

stochastic environment.

Many real-world stochastic environments, however, are not composed of independently-varying components. Information about the change in one component can be used to predict caused or correlated changes in another. The description of these relationships is the goal of empirical sciences, and may be considered in this example to be a statistical extension of the cognitive model. Yet it is often not the individual components or even their relationships which concern us, but their collective mean states over time, which may be termed the "pattern" of the environment. Whereas the cognitive model may perform "well enough" on a component-sampling basis for pattern tracking, the genetic model actually outperforms it when the pattern is completely consistent over even a few transitions (note the results in Part II). At the same time, the genetic model does not discard sources of new schemata when an optimum is obtained, allowing for recovery over another set of absorbing transitions when the pattern is altered in a realistic, correlated fashion. In contrast, it is possible for a component-sampling model to lock into a present optimum that was maintained over sufficient transitions: the probability for change would become miniscule. In particular, note the results of the Part II lag tests -- the initial performance level was maintained with very little variation.

One area of current software development for which these findings have special significance is that of voice recognition and synthesis. Again, the concern is with an overall pattern in the collective mean states of phonemes, not in their individual variation. A population of structures consisting of variations in specific phoneme enunciation (alleles) might be maintained and genetically manipulated to quickly and accurately match incoming phonemic constructs. A wider range of enunciation variability could be tolerated with a level of accuracy at least as good as individual component sampling at much less the computational overhead.

There are also implications in the field of population and evolutionary biology for these findings. It has recently been postulated that evolution occurs in spurts, rather than with a steady progression (1). "Missing links" have been found for very few species. As can be discerned from the lag data below for the genetic group (BETA) large "jumps" in performance change occur over very small transition increments. A steady state is then achieved, with another "jump" then occurring.

RUN 1		
Lag No.	t	Pr /t/
1	1.16	.2518
3	1.57	.1226
7	1.66	.1036
jump [8	1.73	.0906
10	1.97	.0563
11	1.94	.0593
12	1.97	.0567

RUN 1 (continued)

	Lag No.	t	Pr /t/
jump [13	1.91	.0644
	14	1.94	.0607
	15	2.06	.0467

RUN 2

	Lag No.	t	Pr /t/
jump [1	.91	.3696
	3	1.73	.0907
jump [5	2.14	.0382

A caveat must be issued concerning the high level of abstraction of this model and its use of the simplest, and by no means only, genetic operators. However, observance of such a sequence of change even at this level is significant and warrants further refinement and testing of the model.

Secondly, most species in a consistent environment evolve toward "specialization", in which they occupy a very narrow niche. Catastrophic perturbations (as in the fluctuating environments of Part I in which components are often inversed) doom the most highly specialized. In the above genetic model, information processing and integration into the population base was too slow to track such an environment above mediocre performance. In a consistent environment, however, the genetic model evolved toward and maintained a set of highly similar, suitably matching structures. Random point mutation proved fruitless in all instances. This model may thus provide the basis for a suitable abstraction of population evolution and niche occupation.

In his book, Holland lists numerous other areas of possible application, as well as specific, concrete models that have been developed based on genetic operators. It is the authors' hope that the presentation of software and empirical data supporting Holland's model may motivate further interest in this area.

REFERENCES

- (1) Gould, Stephen Jay, Ontogeny and Phylogeny, Cambridge, Mass.: Harvard University Press, 1977, 501 pp.
- (2) Holland, John H., "Adaptation," in Progress in Theoretical Biology, V 4, ed. by R. Rosen and F. M. Snell, NY: Academic Press, 1976, pp. 263-293.
- (3) Holland, John H., Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, Ann Arbor: The University of Michigan Press, 1975, 183 pp.
- (4) Quirin, William L., Probability and Statistics, New York, Harper & Row, 1978, p. 281