Evolutionary Programming

Artificial Intelligence Through Simulated Evolution

Introduction:evolution vs intelligence

- Life on earth has evolved for some **3.5 billion years.**
- Initially only the **strongest creatures** survived, but over time some creatures developed the ability to recall past series of events and apply that knowledge towards making **intelligent** decisions.
- The very existence of humans is testimony to the fact that our ancestors were able to **outwit**, rather than **out power**, those whom they were in competition with.
- This could be regarded as the beginning of **intelligent behavior.**

Theses:

1. Evolution leads to intelligent natural systems

2. Evolutionary methods can lead to realization of intelligent artificial systems

Picture provided courtesy of www.dinodon.com

Introduction:intellectual adaptation

- Some species were able to compete in the survival game by having an increased number of offspring,
- Others survived through making themselves well hidden by making use of camouflage
- We will focus our attention on those “creatures” whose response to the threat of their environment was ***intellectual adaptation.***

Picture courtesy of www.dinodon.com

Introduction

- ***Simulated evolution*** is the process of **duplicating** **certain aspects** of the evolutionary system
- Cellular Automata and Partial Differential Equations are most general models of all phenomena and systems in nature.
- With Simulated Evolution we will produce artificially intelligent automata
  - capable of *solving problems* in new and undiscovered ways.
- We hope to discover a deeper ***understanding of the very organization of intellect.***
- The basis of this approach:
  - humans appear to be very intelligent creatures,
  - there is no reason to believe that we are the most intelligent creatures that could possibly exist.

Table of contents

Theory

"Intelligent behavior is a ***composite ability*** to predict one's environment coupled with a translation of each prediction into a suitable response in light of some objective"

(**Fogel et al., 1966, p. 11**)

Success in **predicting an environment** is a prerequisite for **intelligent behavior.**

Theory

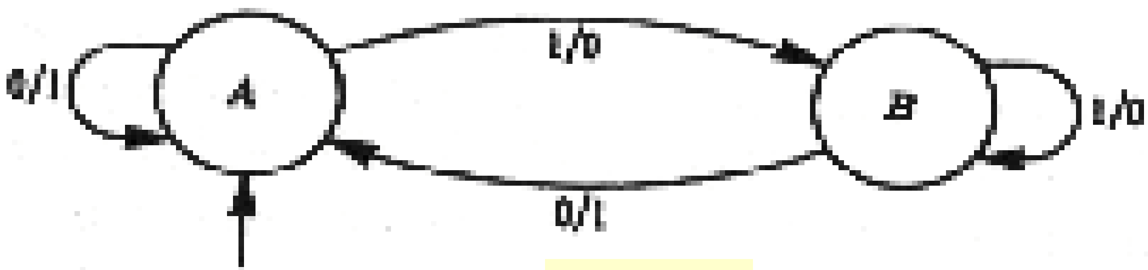
- Let us consider the **environment** to be a sequence of symbols taken from a finite alphabet.
- In general, environment is a state machine and we are a state machine that wants to predict the symbol from the environment
- The task before us is to create an algorithm that would:
  - operate on the observed *indexed set of symbols* and
  - ***produce an output symbol*** that agrees with the next symbol to emerge from the environment.

•If we know the environment, this would be easy problem for FSM theory

•But we do not know the environment, we want to learn it

Theory

This is input sequence of the machine



This is what we see at the output of our machine

To B

Environment:	1	0	1	1	1	0	0	1	1	1	0	1
FSM Predicts:		0	1	0	0	0	1	1	0	0	0	1
Current State:		A	B	A	B	B	B	A	A	B	B	B	A

Agreements in 6 cases

Disagreements in 5 cases

Input sequence 10 leads machine to state A

Theory

- **The basic procedure is as follows:**
- A collection of algorithms makes up the initial population.
- They are **graded** based on how well they predict the next symbol to come out after being fed the given environment.
- The ones that receive a grade above some threshold level are retained as parents for the next iteration, the rest are discarded.
- These offspring are then judged by the same criteria as their parents.
- The process continues
  - until an algorithm of sufficient quality is achieved
  - or the given time lapse period expires.

Theory

- The machines can be judged in a variety of ways:
  - whether or not it predicted the next symbol correctly, ***one at a time***, or
  - we could first *expose the machine* to a number of symbols taken from the environment, **then let it guess.**
- **Addition:** maintain efficiency by penalizing complex machines.

- DEFINITION:
  - **recall length:** how many symbols we expose the machine to before *it has to make* it's prediction.

1.2 Prediction Experiments of Fogel

Prediction Experiments

- In Fogel's Prediction Experiments, there is a given environment at the start
- It is a series of symbols from the input alphabet.
- The initial machines are all identical.
- They are run through the environment and judged based on how well they predict the symbols that follow.
- At the end the best three machines are **kept** and run through a series of mutations to create 3 more offspring.
- **All 6 machines** are then run through the same testing procedure, and the best 3 are chosen… and so on… {(P + C) selection.}

- Every five iterations, **the best machine** is taken and told to predict the next symbol based on the last input symbol given,
  - and the output given is taken and *attached to the environment string*.

Prediction Experiments

The Fogel experiments were done using the 5-state machine in Table 1.1 as the initial machine (all of the seed machines were a copy of this one).

State	Input Symbol	Next State	Output Symbol
1	1	2	1
1	0	1	1
2	1	3	1
2	0	2	0
3	1	4	1
3	0	5	0
4	1	5	1
4	0	3	0
5	1	1	1
5	0	2	0

Table 1.1

Prediction Experiments:Sensitivity to mutations

- The first **four experiments.**
- **Goal:** to demonstrate the sensitivity of the *procedure's capability* to **predic**t symbols in the sequence as a function of the types of mutation that were imposed on the parent machines.
- The environment used was the repeating pattern **(101110011101)*.**
- Only a *single mutation* was applied to each parent to derive it's offspring.

Prediction Experiments

- The first **four experiments.**
- These initial experiments have *no penalty* for complexity.
- **Consider,** why a penalty for complexity?
  - Because huge machines would simply develop that are nothing but the sequence of symbols we input!
- This is **not** the desired end!
- Occam's Razor, overfitting.
- The Experiments 1, 2, 3, and 4 differed in mutations

We will evaluate the **quality of learning** and the ***size of the machine***

Prediction Experiments

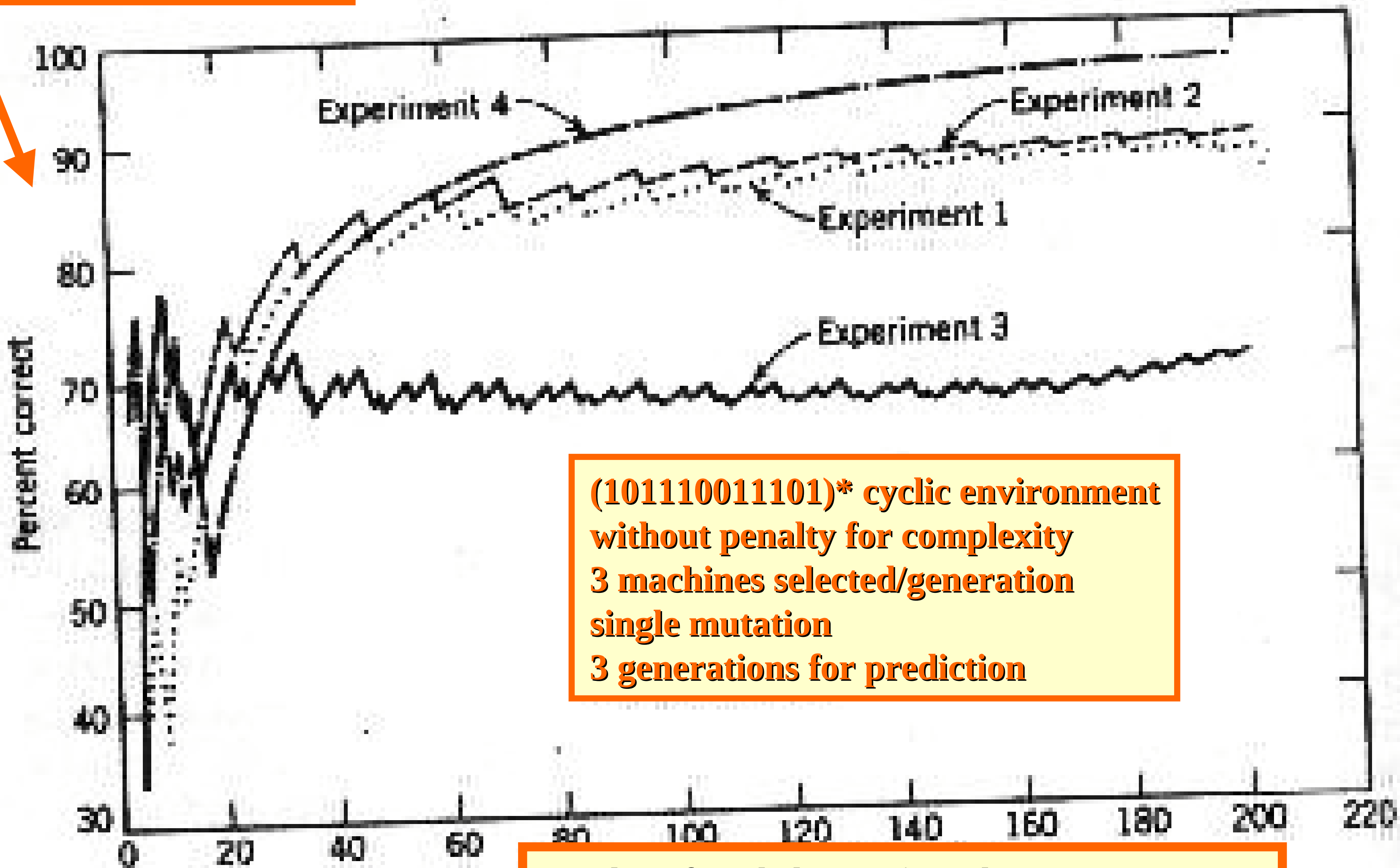
- ***Mutation was one of these 5 types :***
  - Add a state (with certain probability pi)
  - Delete a state (with probability pj)
  - Randomly change a next state link
  - Randomly change the start state
  - Change the start state to the 2nd state assumed under available experience.

All mutation types specified by certain probabilities

Prediction Experiments

Figure 1.5

Percent Correct



(101110011101)* cyclic environment
without penalty for complexity
3 machines selected/generation
single mutation
3 generations for prediction

Number of symbols experienced

Prediction Experiments

- *Several thousand generations* were undertaken.
- each of the final machines grew to between 8 and 10 states.

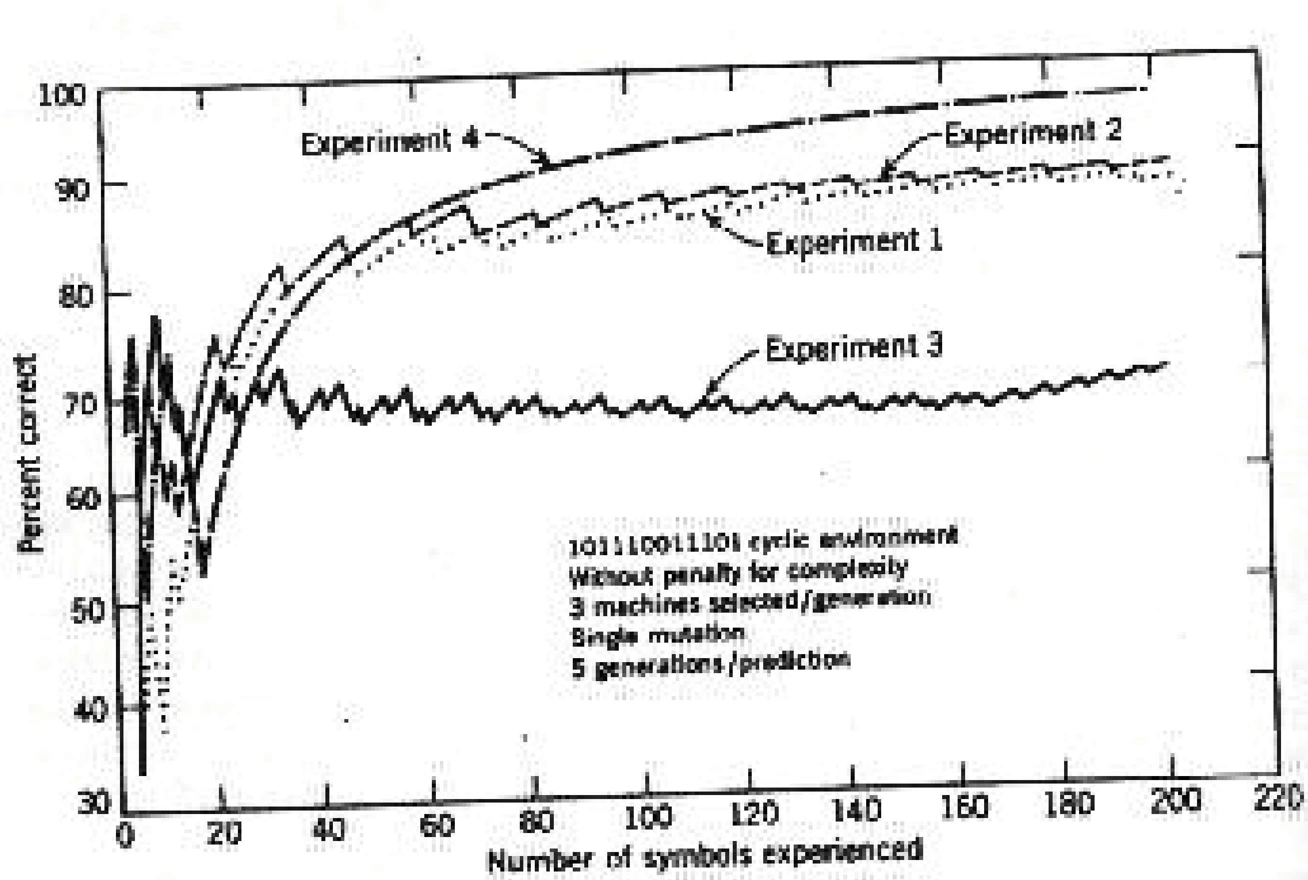


Figure 1.5

Prediction Experiments

- In **Experiment 4** a series of perfect predictor-machines were found after the 19th symbol of experience.
- Poorest prediction occurred in experiment 3, but even this machine showed a remarkable tendency to predict well after the first few iterations of the environment string.
- The **1st experiment** is considered typical and will be *used as the basis* for comparison from now on.

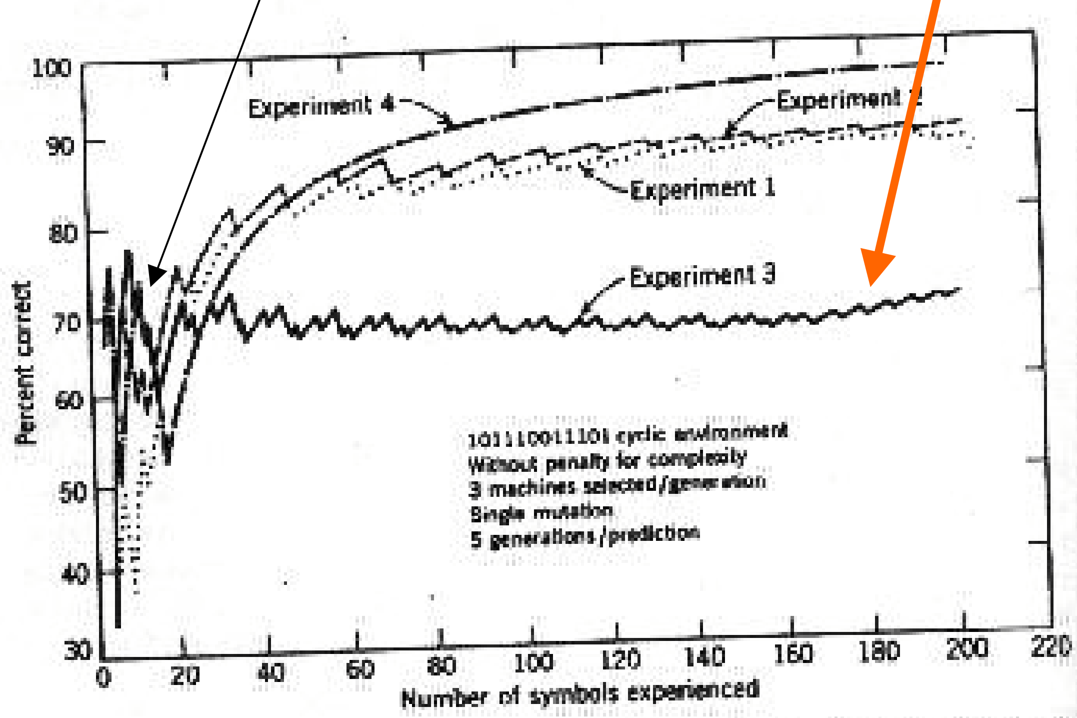


Figure 1.5

Prediction Experiments
Machine Complexity

- The effect of imposing a penalty for machine complexity is shown in figure 1.6.
- The solid curve of experiment 5 represents experiment 1 duplicated with a penalty of 0.01 (or 1%) per state.

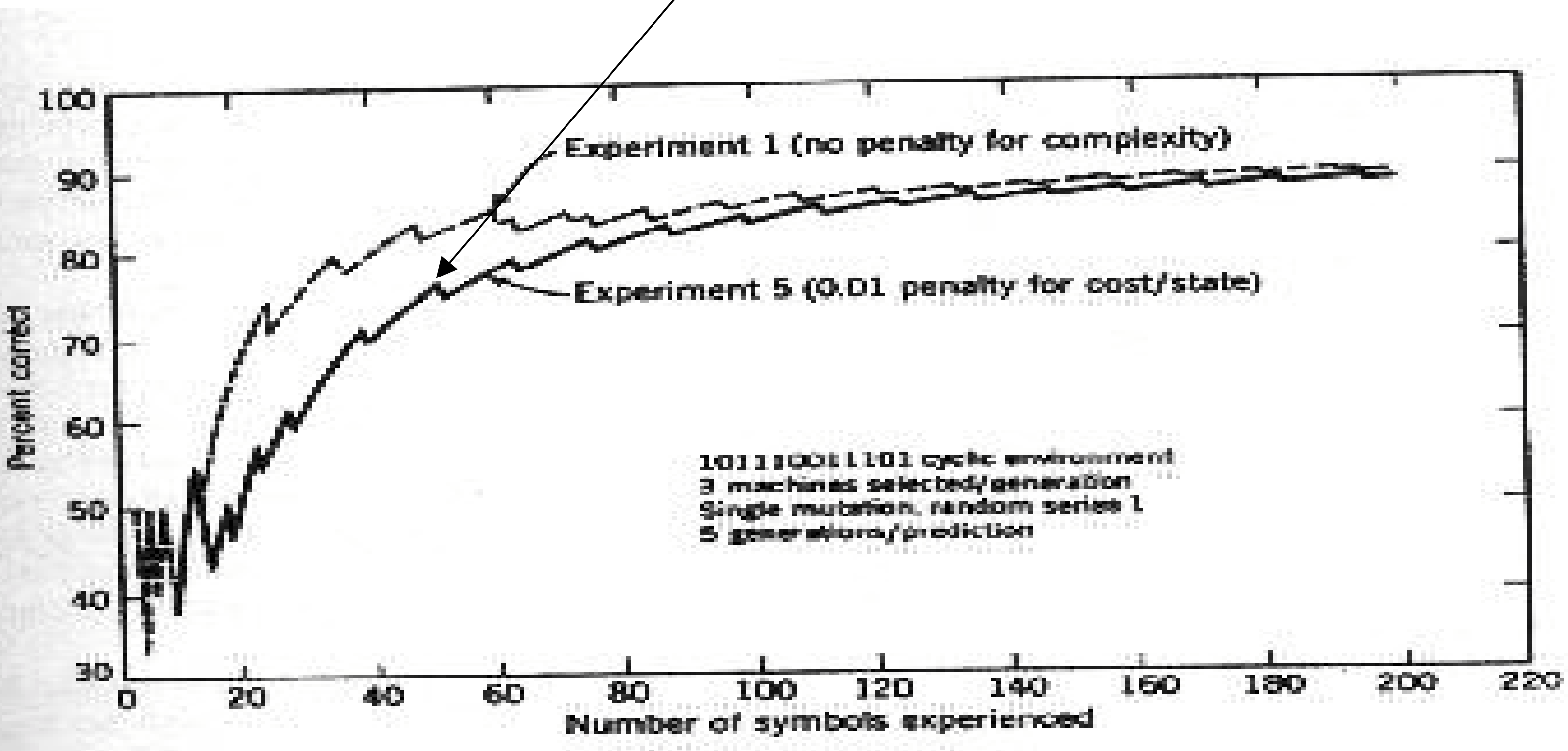


Figure 1.6

Prediction Experiments

Machine Complexity without or with penalty

- The benefit penalty for states is in figure 1.7,
- experiment 5 has significantly less states of FSM,

Number of states of FSM learned

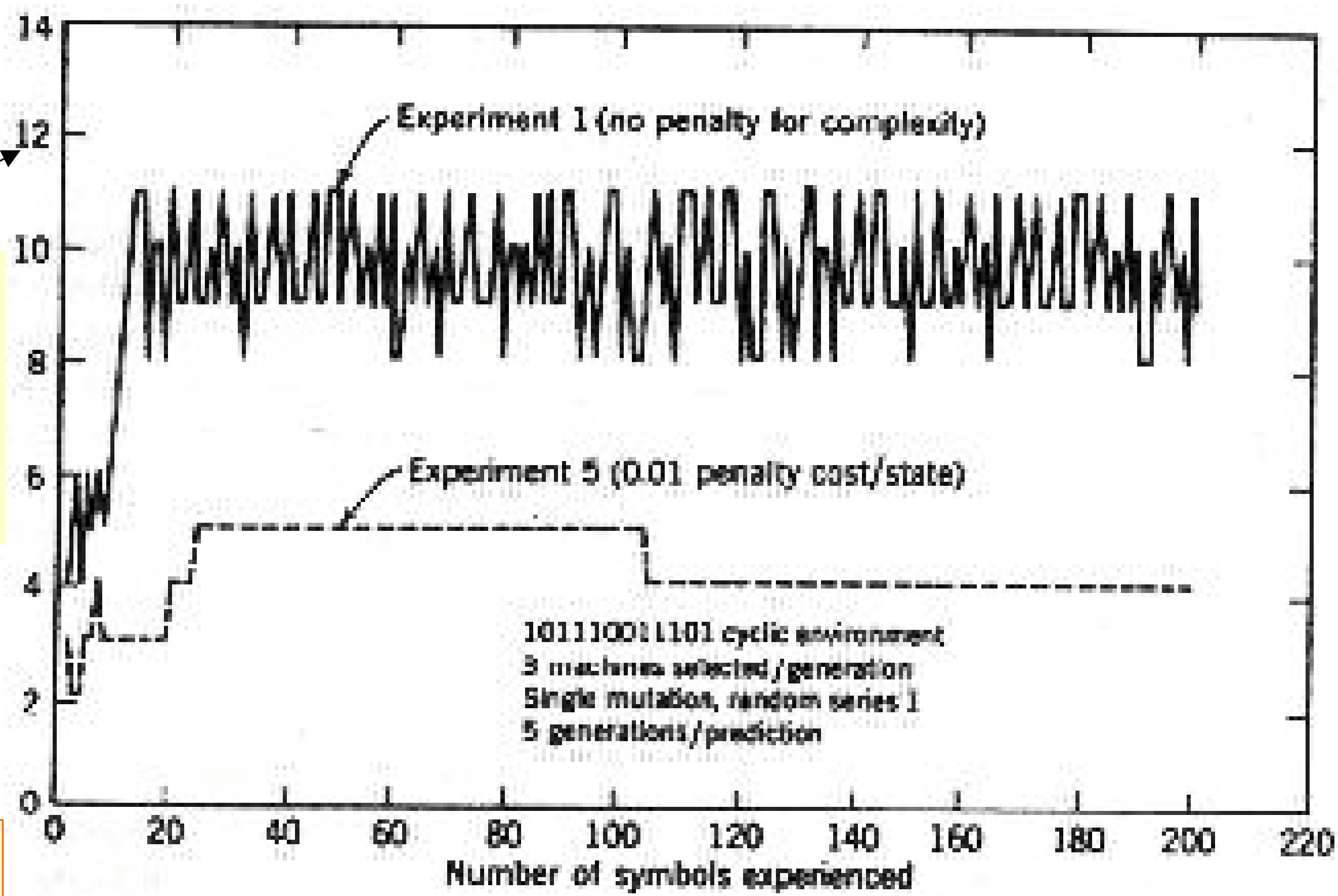


Figure 1.7

Prediction Experiments

Machine Complexity

- but as we can see in figure 1.6 the only time there is a significant difference in prediction capability is in the beginning.
- So, for long sequences, it is better to have less states.

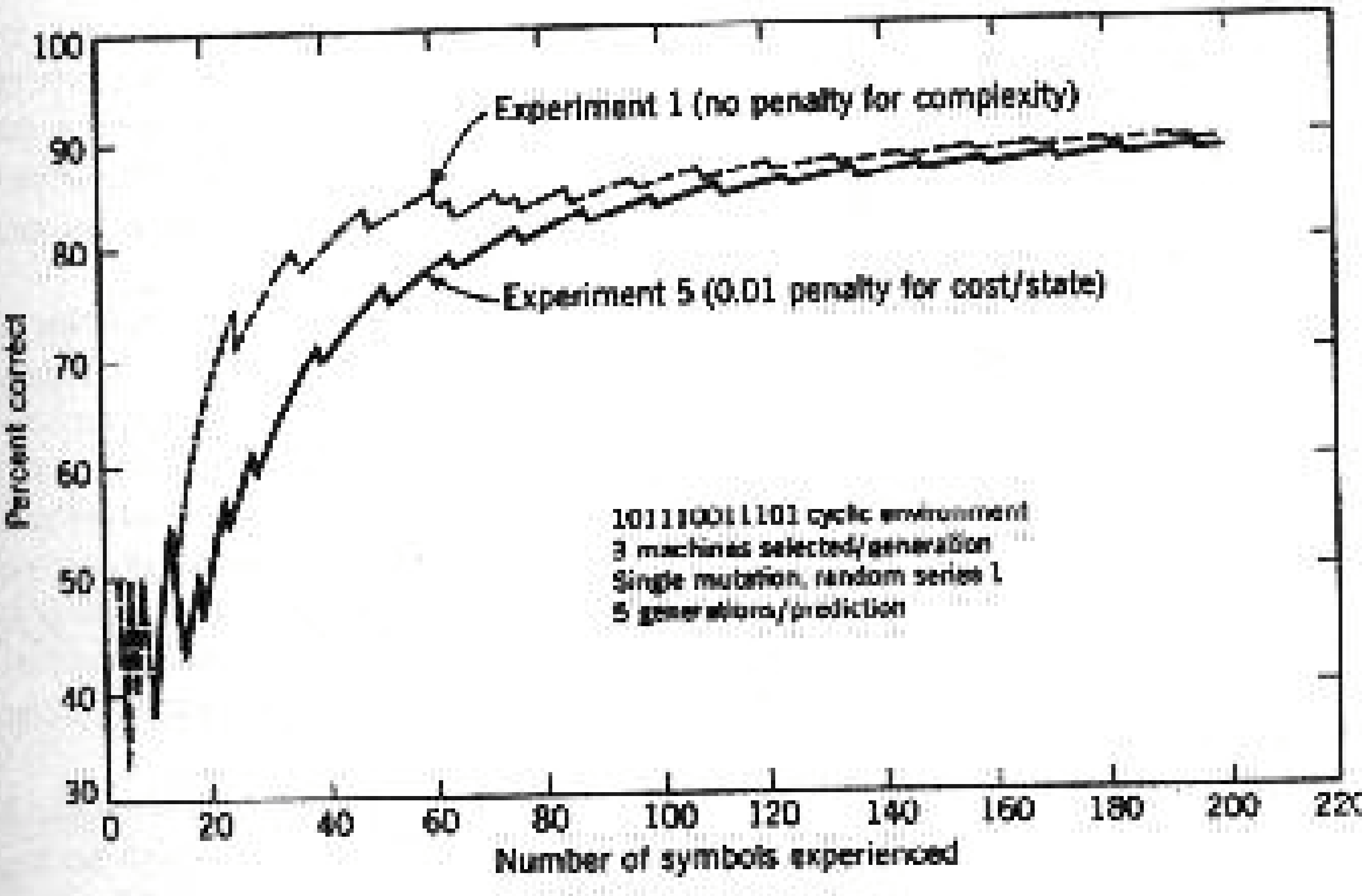


Figure 1.6

Prediction Experiments

Mutation Adjustments

- It is reasonable to suspect that by increasing the probability of the 'add-a-state' mutation we might improve the prediction capability.
- This is demonstrated in figure 1.9, where experiment 6 is a repetition of experiment 1
  - with the probability of the 'add-a-state' increased to 0.3
  - with the 'delete-a-state' probability decreased to 0.1.
- We can see that experiment 6 outperforms experiment 1.
-

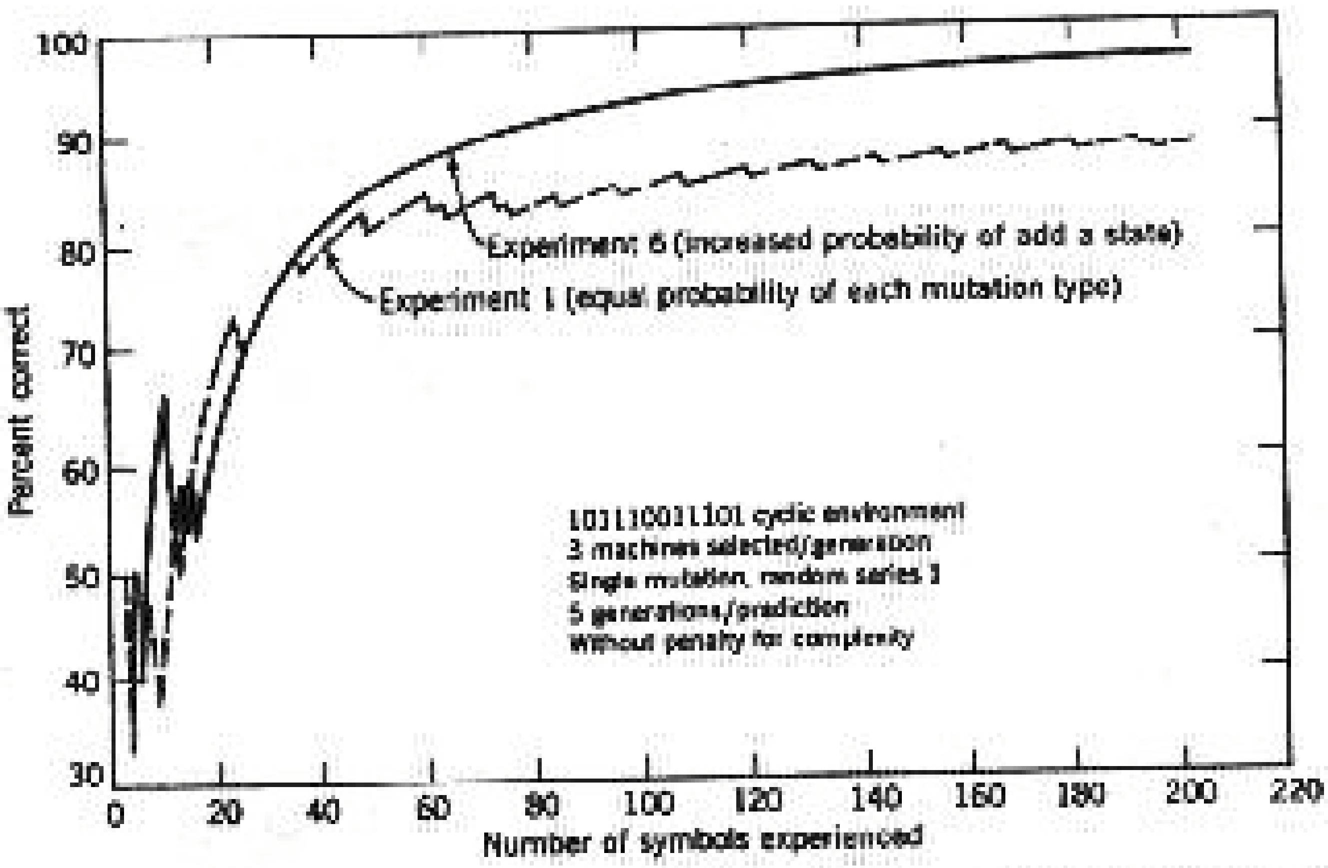


Figure 1.9

Prediction Experiments

Number of Mutations per iteration

- The benefits of increasing the number of mutations per iteration is shown in figure 1.10, which shows experiments 1, 7, and 8 representing single, double, and triple mutation respectfully.
- The size of each of these machines is shown in figure 1.11.

Figure 1.10

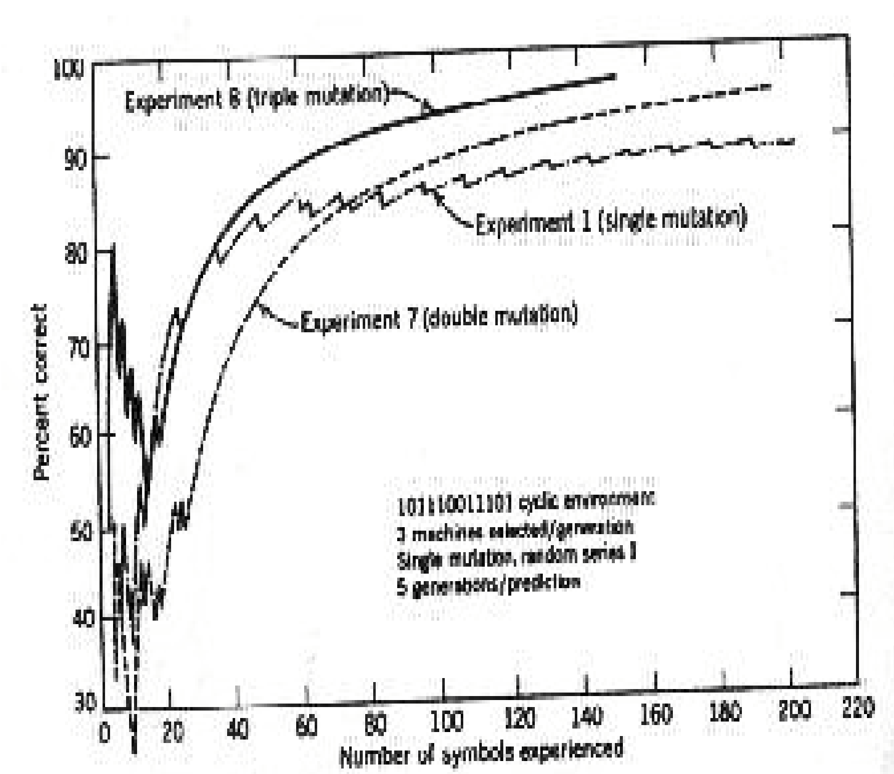
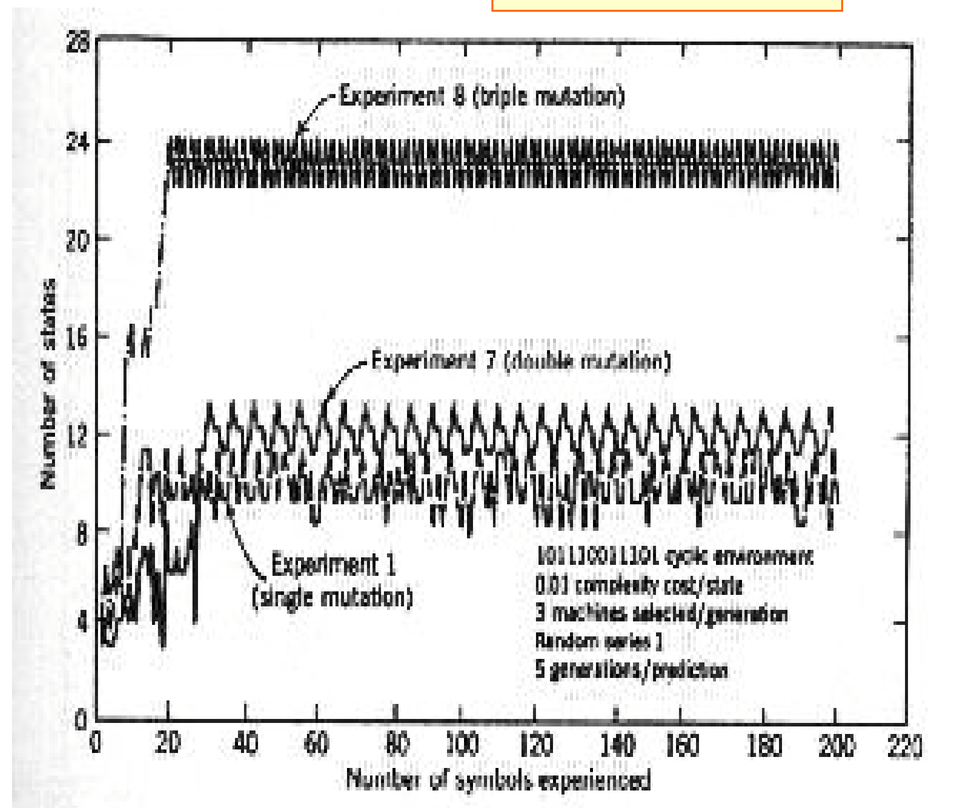


Figure 1.11



Prediction Experiments

Recall Length in stable and noisy environments

- In the case of a purely cyclic environment with no change to the input symbols, increasing the recall length provides for a larger sample size and an ***increased prediction rate.***

- In a noisy environment that has changes to the environment string it might be better to forget some past symbols

Prediction Experiments

Role of Recall Length in Stable (Cyclic) Environment

- Figure 1.12 shows the difference in recall lengths.
- During the initial sequence, the behavior appears quite random, but one can see that the longer recall length did exhibit **faster learning of the cyclic environment.**

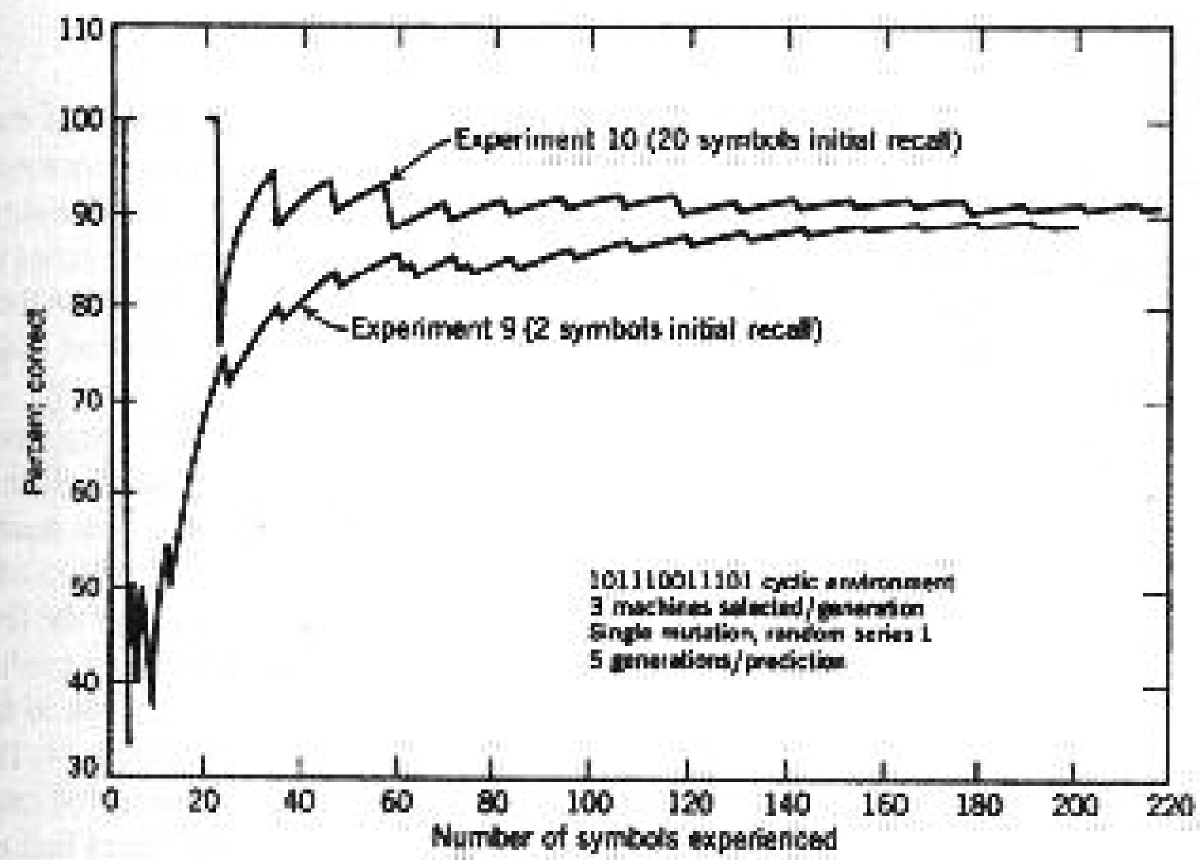
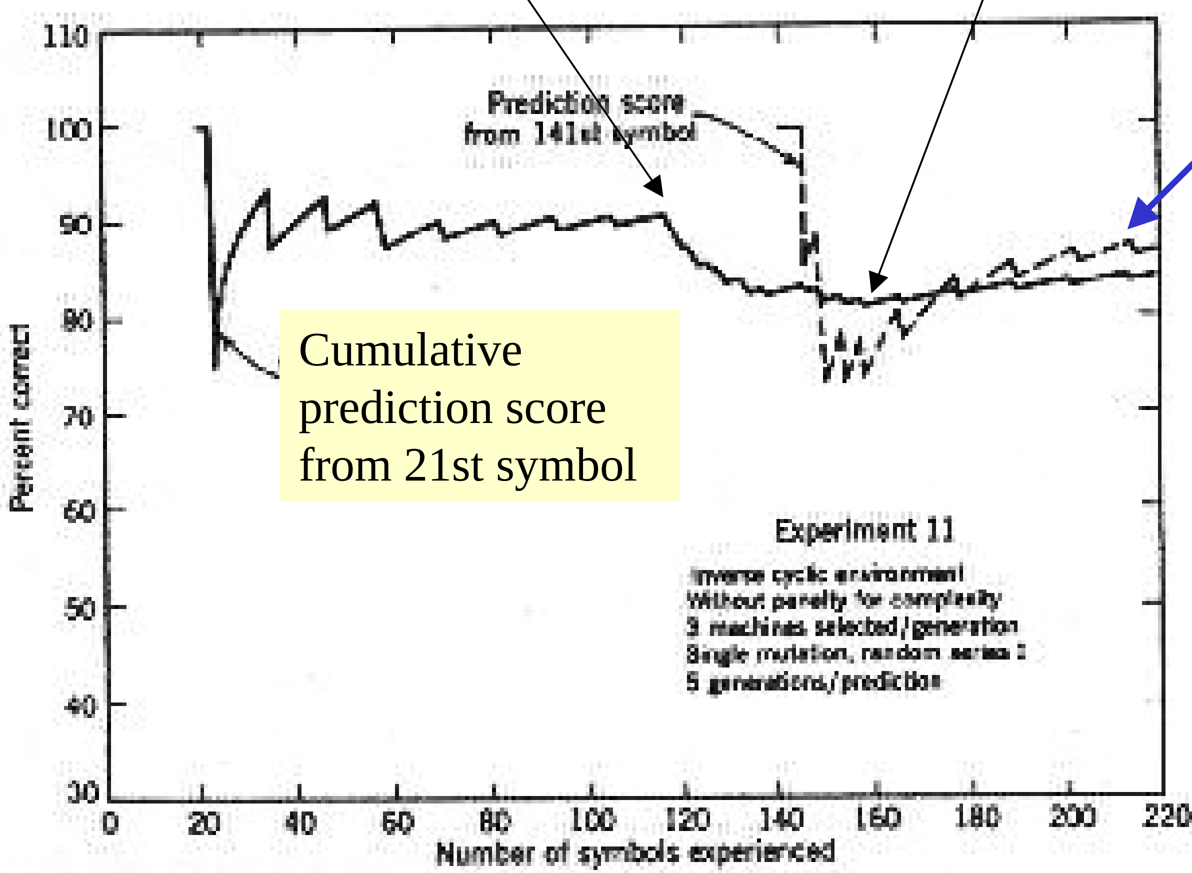


Figure 1.12

Prediction Experiments

Radical Change in Environment

- Figure 1.13 and 1.14 demonstrate some interesting behavior.
- The solid line of Figure 1.13 demonstrates a normal evolutionary transition, but at symbol number 120 the environment undergoes a radical change.
- This change was the complete reversal of all the symbols in our environment.



Shorter sequence quicker adapts to new environment

Figure 1.13

Is this “simpler animals quicker adapt to new environment so dinosaurs died?

Prediction Experiments

Radical Change in Environment

I was at this point that the number of states increased quickly as a great deal of "unlearning" had to take place.

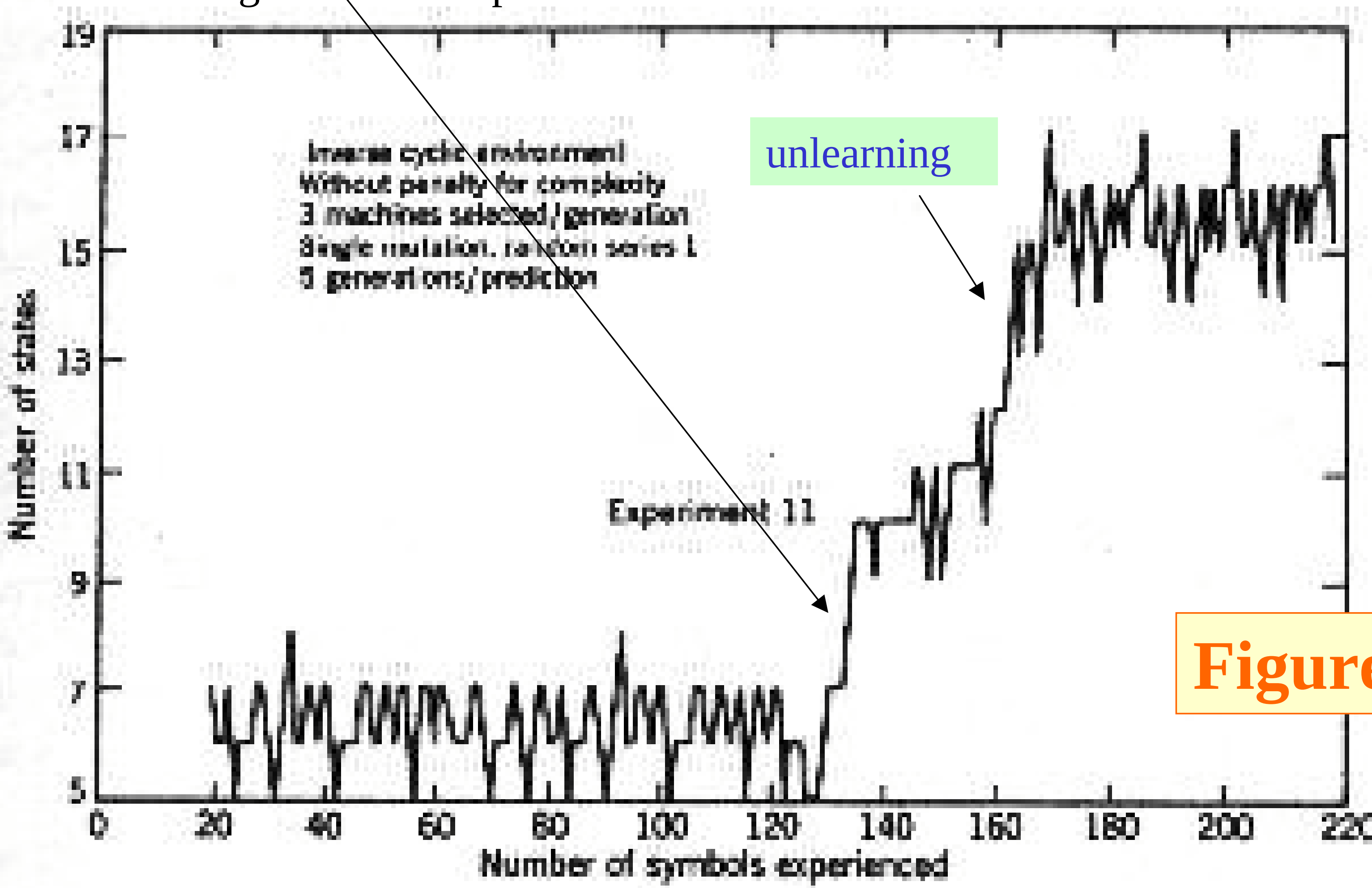


Figure 1.14

Prediction Experiments

Radical Change in Environment

- The dotted line in figure 1.13 shows the comparison of machines that were not exposed to the radical change and instead started after it had already occurred.
- This score compares favorably with the first solid line when one considers that a machine is judged over the entire length of it's experience.

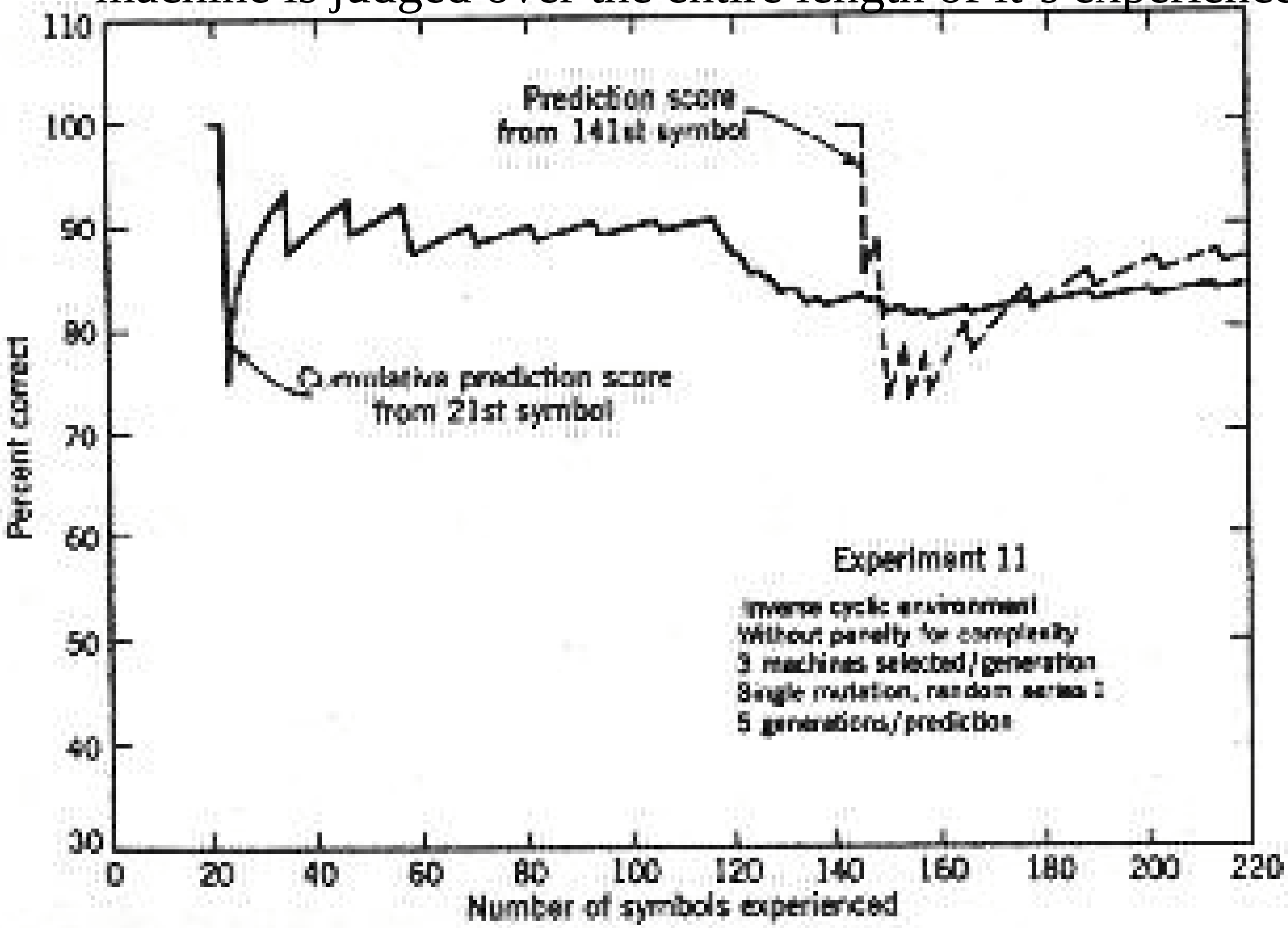


Figure 1.13

New Series of Prediction Experiments

Predicting Primes

- The new interesting experiments:
  - the environment represents prime numbers in an incremental count within the string.
- For example, 01101010001, digits 2, 3, 5, 7, and 11 are all 1's
  - which are all the prime numbers.

2

11

01101010001

Prediction Experiments

Predicting Primes

- We can see in figure 1.16 that experiment 15 ended up predicting the prime numbers quite well towards the end, and we can see in figure 1.17 that it ended up with very few states.
- This is easily understood when one notices that the higher we get into the environment string the less frequent prime numbers become.

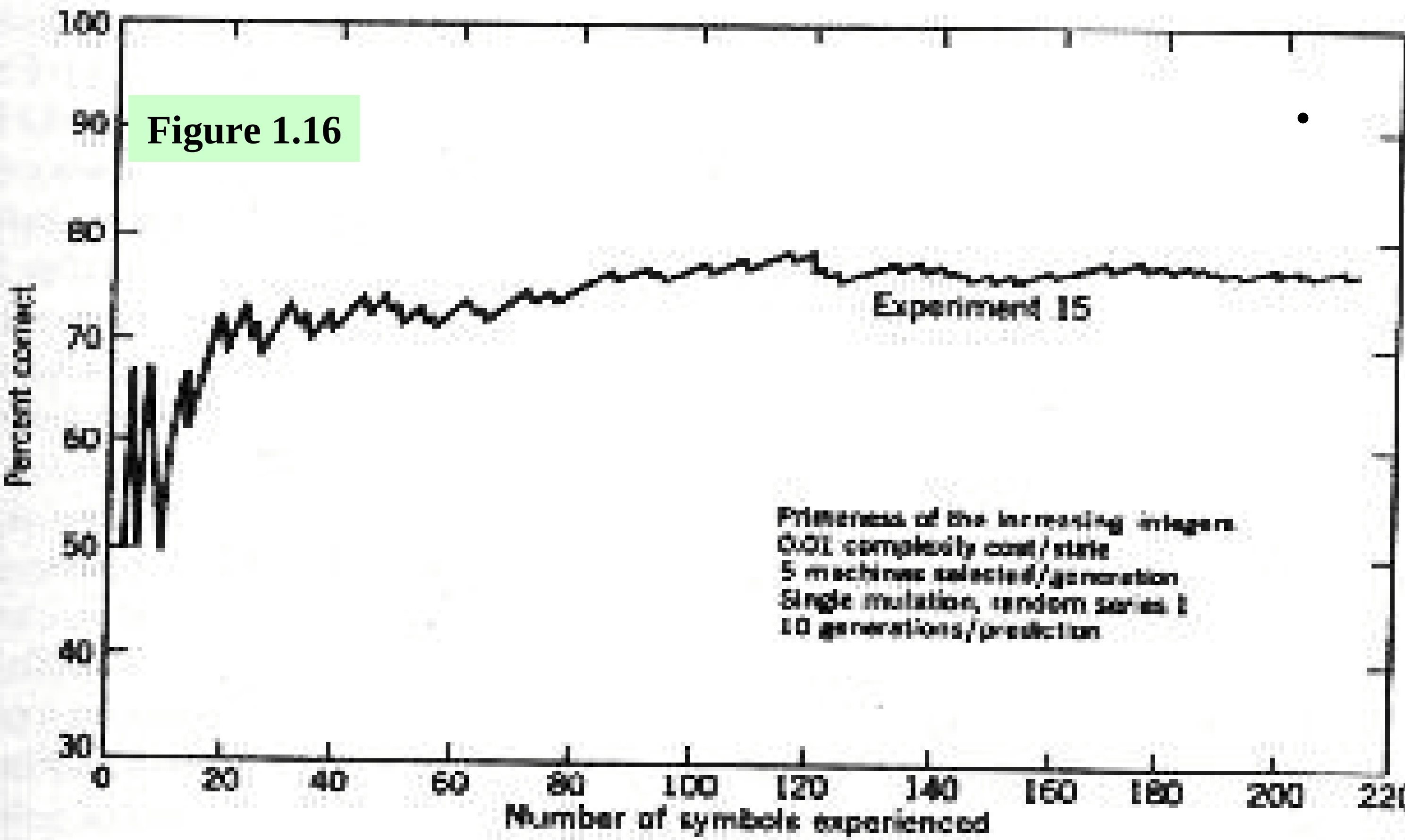


Figure 1.16

- The results were obtained with a penalty for complexity of 0.01 per state, 5 machines per evolutionary iteration, and 10 rounds of mutation/selection before each prediction of a new symbol.

Prediction Experiments

Predicting Primes

- We can see in figure 1.17 that Experiment 15 ended up with **very few** states.
- This is easily understood when one notices that the higher we get into the environment string the less frequent prime numbers become.

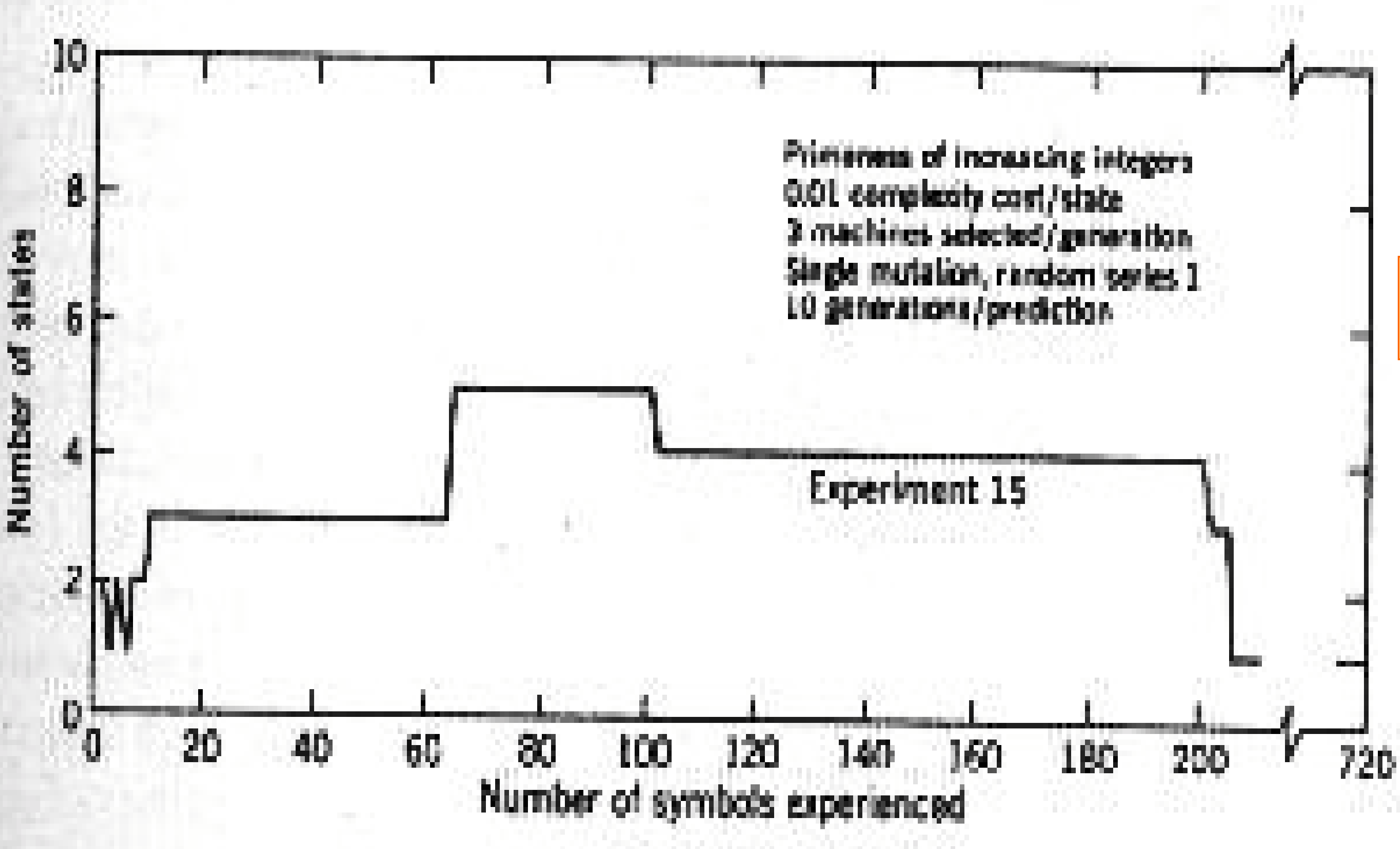


Figure 1.17

Prediction Experiments
Predicting Primes

- To make things more interesting Fogel et al <u>increased the length</u> of recall and gave a <u>bonus for predicting a rare event.</u>
- So the score given for predicting a 1 was the number of 0's that preceded it and the score given for predicting a 0 was the number of 1's that preceded it.
- One can see that predicting a 1 is <u>much more valuable</u> than predicting a 0.
- Analysis of the results showed that the machines quickly "learned":
  - to **recognize numbers divisible by 2 and 3 as not prime**,
  - some hints towards an increased tendency to predict multiples of 5's as not prime.

Experiment with Humans

- *Experiment:*
  - human subjects were given a recall frame of 10 and asked to predict the next symbol
- ***Result.***
  - The evolutionary process consistently outperformed **the humans.**
- One may argue that this is unfair :
  - on one side machines adapting through several iterations
  - on the other humans who are unchanging,
- But at this point we regard the system itself as the intelligent process,
  - not just the single iteration of a machine.

Is FSM better?

They key is ***Adaptation***

- The key to the success of the evolutionary machines is in their continual adaptation to the environment.
- The goal is not to ***end up*** with a final machine that can predict well.
- The goal is to come up with a process that through continued mutation/selection **always** generates the best machine.

Evolutionary Programming is Evolution of FSM

- **Evolutionary programming** is not so much about programming, its more about the **evolution** of automaton.
- Compared to the ***genetic algorithms***:
  - you don't just have a bit string that encodes parameters,
  - you have to encode the initial state, the transition table, and the alphabet,
  - then you have to come up with problem specific mutations, or genetic mutators…
- There is nothing like the *recombination mutation*

1.3 Pattern Recognition and Classification

Pattern Recognition and Classification

- The key to understanding a sequence of foreign symbols is to try and find a recognizable pattern within them.
- If there is no pattern, it is assumed to be random.
- *In contrast:* if we can turn out a good prediction score it may reveal the presence of an unchanging signal.
- ***Variability in prediction score*** means the data may contain a message.
- If we CAN demonstrate a good prediction score, the question arises:
  - what is the nature of the signal?
- The state machine with acceptable score is a *good description in itself*.

Pattern Recognition and Classification

Questions

- How well do these state machines describe the signal?
- How well can they emulate human thought?
- Can they recognize and classify patterns in the same manner as a human operator?

Pattern Recognition and Classification

- **Experiment**
  - A series of broadband signals were generated
  - then expressed in an 8-symbol alphabet,
  - input into a computer program that would evolve to predict their behavior.
- They were generated with the goal of creating 4 sets of 4 signals that held basic similarities,
  - such as the number of peaks and valleys and their locations being roughly the same.

Figure 1.20. 16 patterns for pattern recognition experiment

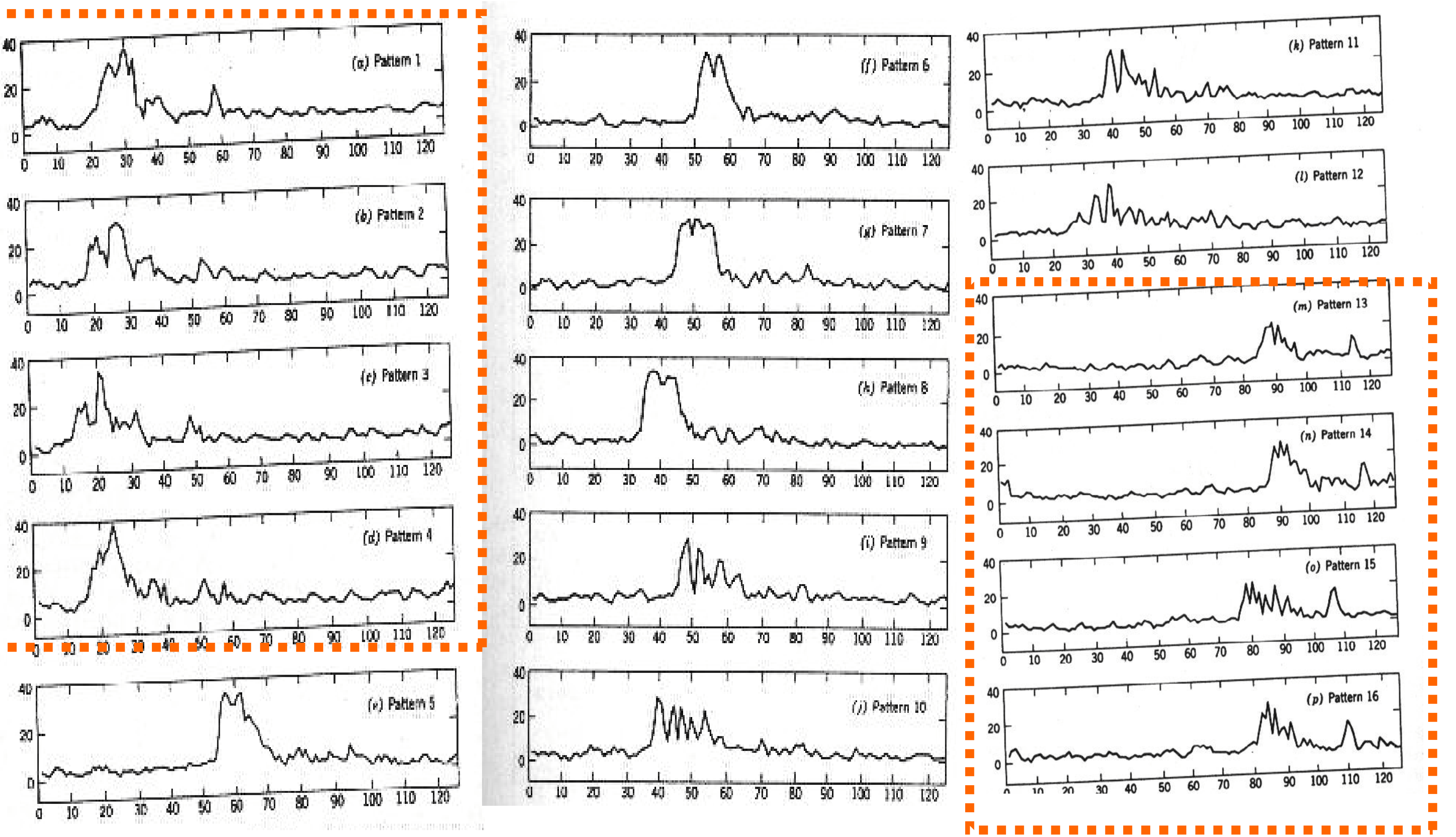
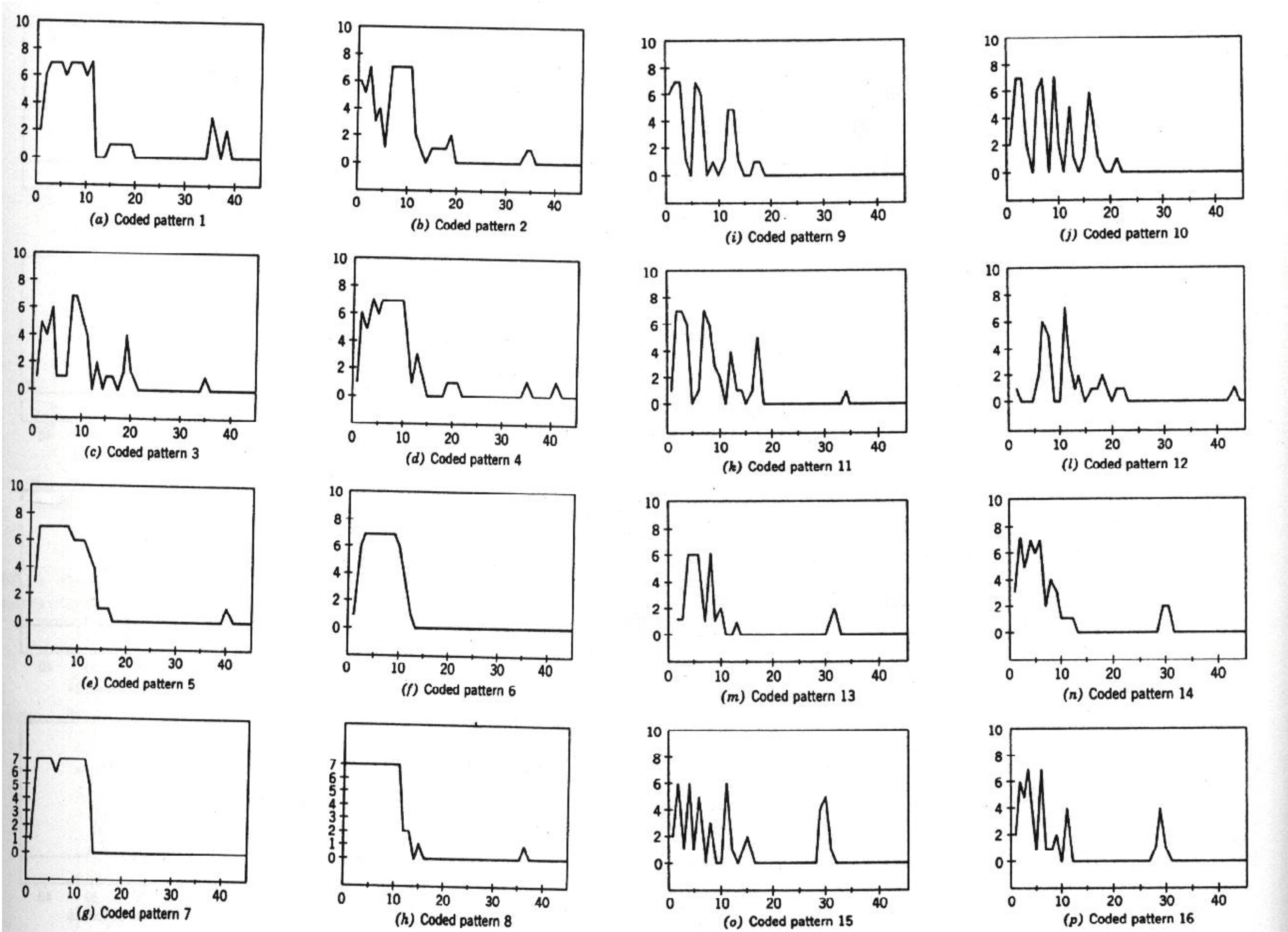


Figure 1.21


(a) Coded pattern 1
(b) Coded pattern 2
(i) Coded pattern 9
(j) Coded pattern 10
(c) Coded pattern 3
(d) Coded pattern 4
(k) Coded pattern 11
(l) Coded pattern 12
(e) Coded pattern 5
(f) Coded pattern 6
(m) Coded pattern 13
(n) Coded pattern 14
(g) Coded pattern 7
(h) Coded pattern 8
(o) Coded pattern 15
(p) Coded pattern 16
0
10
20
30
40

Pattern Recognition and Classification: assumptions

- An eight-symbol evolutionary program was used to predict each next symbol in an unending repetition of each of these patterns.
- There was no penalty for complexity.
- There was10 generations prior to each prediction.
- There was also a "magnitude of the difference" *error cost matrix specification* of the goal.

Pattern Recognition and Classification

- Table 1.2 indicates the average prediction error rate of these evolutionary programs applied to their own signal after the first 50, 100, 200, and 400 predictions.
- It can be seen that the greatest amount of "learning" occurred in the *early stages of development.*

	Recall Length			
Pattern	50	100	200	400
1	0.330	0.172	0.158	0.165
2	0.307	0.306	0.211	0.145
3	0.228	0.195	0.091	0.056
4	0.248	0.209	0.111	0.114
5	0.106	0.091	0.057	0.045
6	0.043	0.051	0.051	0.047
7	0.082	0.041	0.048	0.050
8	0.268	0.267	0.288	0.276
9	0.185	0.163	0.172	0.149
10	0.207	0.141	0.088	0.070
11	0.184	0.082	0.078	0.052
12	0.312	0.107	0.090	0.072
13	0.122	0.102	0.091	0.081
14	0.103	0.126	0.110	0.109
15	0.165	0.176	0.152	0.136
16	0.063	0.068	0.067	0.065

Pattern Recognition and Classification: what is similarity of patterns?

- Each evolved machine was a characterization of the signal in which it developed, this is obvious.
- One might think it is also obvious that we ***recognize similarities*** in the signals through similarities in the machines,
  - but this is not such an easy task since these machines can often grow to be very complex,
  - and what method would you use to make such a comparison?
- It is much more natural to accomplish the comparison by allowing the evolved machines to *attempt a prediction of the OTHER, similar signals.*
- The ***similarity between patterns*** should be demonstrated by the **similarity in prediction scores.**

Pattern Recognition and Classification

- Table 1.3 shows the results of such a comparison:
  - Things did not turn out the way we had hoped.
- As was expected, each machine predicted it's own signal very well.
  - but the remaining scores showed that none could classify the signals in the desired manner.

	Predictor-Machine 1 Evaluated over		Predictor-Machine 2 Evaluated over		Predictor-Machine 3 Evaluated over	
Pattern	45 Symbols	90 Symbols	45 Symbols	90 Symbols	45 Symbols	90 Symbols
1	0.250	0.292	4.727	5.865	12.136	12.045
2	4.727	5.315	0.023	0.427	5.159	7.787
3	4.841	4.697	4.864	4.820	0.023	0.034
4	4.500	3.629	5.659	6.00	11.159	11.258
5	2.796	3.348	5.886	5.112	5.432	8.843
6	1.091	1.539	3.136	3.933	9.068	9.551
7	1.296	1.697	3.364	3.112	11.886	11.933
8	1.432	2.753	2.318	2.292	12.091	12.506
9	3.773	4.685	5.000	5.348	4.682	5.315
10	6.273	6.787	8.546	7.225	6.068	6.472
11	5.250	5.607	5.136	4.798	3.818	4.292
12	3.000	2.978	3.250	3.225	4.727	4.573
13	4.432	4.258	3.136	3.337	3.796	3.764
14	5.341	5.382	3.818	4.112	6.727	6.753
15	4.455	4.854	4.386	4.214	5.046	4.989
16	3.682	4.090	2.636	3.416	5.682	5.663

Table 1.3

Pattern Recognition and Classification: humans versus machines

- Predictor machines recognize ***similarity*** differently than humans.
- A *human operator* would simply look at the signals and note the number of peaks and valleys and their relative position and magnitude, making the comparison a trivial task.
- But there is *no demand* that the evolutionary program emulate human behavior in performing the same task.

According to Fogel, it is this very constraint that has limited the advancement of AI in the past 30 years.

Control System Design

- So far we've looked at such problems as:
  - **detection** (Is there a signal?)
  - **discrimination** (if so, what is the signal?),
  - **recognition** (has the signal been seen before?),
  - **classification** (if not, which of a set of signals is it most like).

- But almost all of these are of interest only in that they can help to solve the problem of control.

Control System Design

- So what is this problem of control?
- Let us define a system as a *plant*.
- This could be any system:
  - a computer program,
  - another state machine,
  - a living organism.
- We have no idea what the nature of this system is:
  - all we know is that given some input string it will produce some output string.
- The problem of control = to understand such a system.
- We want to be able to tell the plant what to do
  - and have it achieve some desired result *or goal*.

Control System Design

- But:
- if we don't understand anything about the nature of the system,
- and only have an output that was spewed out by the plant *on some given input*,
  - how can we possibly **hope to be able to control** such a system (be able to *tell it what to do*)?
- We use evolutionary programming.

How do we use evolutionary programming to solve the problem of control?

- The process is as follows :
  - 1. Create a state machine that you believe best describes the plant, but this initial machine is actually not very relevant.
    - In theory, it could be anything, but we should attempt to emulate the plant as close as we can.
  - 2. We then give our newly created machine the <u>sequence of input symbols</u> that was given to our original plant,
    - and judge it based on how well it could predict the actual output that was given by the plant.

Control System Design

3. We continually evolve the machine to become a perfect predictor of the plant,

this meaning that the machine will spit out the same output as the plant when they are both given some input sequence.

4. Now, if we want to control the plant, we need to determine the input string that will achieve our desired end.

To do this we simply look at our state machine and determine the input symbols that would be required to produce our desired output.

Control System Design

- This is where the actual functionality of evolutionary programming comes in.
- It allows us to develop a machine that will further allow us to understand *some unknown system*.

Unrecognized Observations

- There have been several ideas that have been considered as potentially important but were not given sufficient attention because of time and technological restraints.
  - 1. A suitable *choice in mutation noise* may increase the prediction rates of machine.
  - 2. While the best parents will usually produce the best children, lower ranked parents should be retained as protection against gross non-stationarity of the environment (Radical Change).
  - 3. The concept of **recombination** has been quite successful in nature, so perhaps it would be beneficial in evolutionary programming experiments as well.

Summary

- So let's look at the whole thing in perspective.

- Intelligence was defined as:
  - the ability to **predict a given environment**,
  - coupled with the ability to *select a suitable response* in light of of the prediction and the given goal.
- The problem of predicting the next symbol was reduced to the problem of **developing a state machine that could do the same,** given some environment.
- These machines were driven by the available history and were evaluated in terms of the given goal.

Summary

- But we need not constrain ourselves to a symbol predicting machine,
- In fact the same process could be applied to any well defined goal within the constraints of the system.
- Thus the evaluation will take place in terms of response behavior, in which prediction of one's environment is an implicit intervening variable.
- We have seen a variety of such experiments.

Summary

- But even further implications are possible.
- The ***scientific method*** could be regarded as an evolutionary process in which a **succession of models are generated and evaluated**.
- Therefore, simulation of the evolutionary process is tantamount to a **mechanization of the scientific method.**
- **Induction,** a process that previously was regarded as requiring creativity and imagination has now been reduced to a routine procedure.

Summary

- So if we make our desired goal one of self-preservation, such machines may begin to display self-awareness in that they can describe essential features of their survival if so requested.

- **What are goals made of?**
  - **They are made up of the various factors that lead towards self-preservation.**
  - **Only those creatures that can successfully model themselves can alter their sub-goals to support their own survival.**
  - **To succeed their self-image must be in close correspondence to reality.**

- With this knowledge we can hope to achieve a greater understanding of our own intellect, or of even greater significance, to create inanimate machines that accomplish these same tasks.

Sources

Derek Sweet

Gary Fogel