PARAMETER ESTIMATION OF A PROBABILISTIC AUTOMATA MODEL OF DNA MEIOSIS

BY

KUAN CHAN WU

B.S., Chung Cheng Institute of Technology, Taiwan, 1963.

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements of the degree

MASTER OF SCIENCE

Department of Computer Science

Kansas State University
Manhattan, Kansas

1976.

Approved:

Virg Wallentine

Major Professor

TABLE OF CONTENTS

ILLEGIBLE

THE FOLLOWING DOCUMENT (S) IS ILLEGIBLE DUE TO THE PRINTING ON THE ORIGINAL BEING CUT OFF

ILLEGIBLE

Acknowledgement

I wish to express my sincere gratitude to my major professor, Dr. Fred J. Maryanski for his guidance, creative suggestions and encouragement during the writting of this report.

LIST OF TABLES

LIST OF FIGURES

Introduction

Automata theory is the study of the dynamic behavior of discrete parameter information systems. Within this area, we define not only problems that deal with digital computers but also problems associated with such topics as describing the behavior of nerve networks, the representation of the properties of languages etc. (1)

A Probabilistic Automaton is an automaton in which the characteristics of the mapping can be described only in a probabilistic rather than a deterministic manner. (2)

The purpose of this report is to estimate parameters of a probabilistic Finite-state Automaton model to fit as closely as possible. The model DNA (Deoxynucleic Acid) Meiosis, with the sample data taken from experiments conducted by Mortimer (3).

Background

For this study we will need the following basic properties.

I. Formal grammars and probabilistic grammars.

Definition I - 1

Grammar: A formal grammar is an ordered quadruple

$G = (V_n, V_t, X_o, F)$

where

V_n - is a finite collection of variables or nonterminal symbols.

V_t - is a finite alphabet of terminal symbols such that $V_n \cap V_t = \phi$

X_o - is the initial letter (start sample) $X_o \in V_n$.

F - is a finite set of ordered pairs (P, Q) elements (P, Q) of F are called rewriting rules or productions and are written $P \rightarrow Q$, $P \in (V_n \cup V_t)^+$, $Q \in (V_n \cup V_t)^*$

P is the premise of the rule.

Q is the consequence of the rule.

$(V_n \cup V_t)^+ = (V_n \cup V_t)^* - \lambda$

λ is the empty string.

Note:

$(V_n \cup V_t)^*$ means any string of zero or more nonterminal or terminal symbols.

Definition I - 2

For i = 0, 1, 2, 3 a grammar $G = (V_n, V_t, X_o, F)$ is of the type i iff the restrictions (i) on F, as given below, are satisfied.

(0) No restrictions.

(1) Each production in F is of the form $Q_1XQ_2 \rightarrow Q_1PQ_2$, where Q_1 and Q_2 are words over the alphabet $V = V_n \cup V_t$, $X \in V_n$ and P is a nonempty word over V, with the possible exception of the production $X_o \rightarrow \lambda$, where λ is the empty string, whose occurrence in F implies, however, that X_o does not occur on the right side of any production in F. G is called context-sensitive.

(2) Each production in F is of the form $X \rightarrow P$, where $X \in V_n$ and $P \in V^*$. According to type - 2 grammars, a nonterminal X may be rewritten as P, no matter what the letters adjacent to X are, this is, independently of the context. Therefore, type - 2 is context free.

(3) Each production is of one of the two forms $X \rightarrow YP$ or $X \rightarrow P$ where $X, Y \in V_n$ and $P \in V_t$. It is called finite-state or regular grammar.

Example 1.

a. Context sentive grammar

$G = (V_n, V_t, X_o, F)$

where $V_n = \{ S, B, C \}$

$V_t = \{ a, b, c \}$

$X_o = S$

F consists of the following:

$S \rightarrow aSBC$	$aB \rightarrow ab$	$bC \rightarrow bc$
$S \rightarrow aBC$	$bB \rightarrow bb$	$cC \rightarrow cc$
$CB \rightarrow BC$		

b. Context free grammar

$G = (V_n, V_t, X_o, F)$

where $V_n = \{ S, A, B \}$

$V_t = \{ a, b \}$

$X_o = S$

F consists of the following:

$S \rightarrow aB$	$A \rightarrow bAA$	$A \rightarrow aS$
$S \rightarrow bA$	$B \rightarrow b$	$B \rightarrow aBB$
$A \rightarrow a$	$B \rightarrow bS$	

c. Regular grammar

$G = (V_n, V_t, X_o, F)$

where $V_n = \{ S, A, B \}$

$V_t = \{ a, b \}$

$X_o = S$

F consists of the following:

$S \rightarrow aA$	$B \rightarrow bB$	$A \rightarrow bB$
$S \rightarrow bB$	$B \rightarrow b$	$A \rightarrow aS$
$A \rightarrow aA$	$B \rightarrow a$	$S \rightarrow a$

Definition I - 3

Language generated by a grammar G is

$L(G) = \left\{ c \mid c \in V_t^* \quad X_o \xrightarrow{*} c \right\}$

Note:

$X_o \xrightarrow{*} c$ means c is derived from X_o using zero or more rewrite rule substitutions.

Definition I - 4

A probabilistic grammar (P - grammar) is defined by the 5 - tuple.

$G = (V_t, V_n, X_o, F, P)$

where V_t, V_n, X_o, F have the same meaning as the formal grammar. P is the set of probabilisties that are assigned by 1 to 1 mapping to the rules of F.

The ith production of G is assumed to have the form P_i: $X \rightarrow Y$. The term P_i is the rule probability and $0 < P_i \leq 1$.

Definition I - 5

For this discussion only finite-state (regular) grammars are considered. The rewrite rules have the form $X \rightarrow ZY$ or $X \rightarrow Y$ where $X, Z \in V_n$, $Y \in V_t$.

Definition I - 6 Unrestricted probabilistic grammar

A probabilistic grammar is unrestricted if the probability of the application of any rewrite rule depends only on the presence of a premise in a derivation and not upon the previous rule applications.

Definition I - 7 Proper grammar

A probabilistic grammar is proper if $\forall c \in V_n$ with k rewrite rules $P_1, P_2, \ldots\ldots P_k$

$$\sum_{i=1}^{k} P_i = 1$$

Definition I - 8

P - language defined by a P - grammar. Let G be a P - Grammar

$G = (V_n, V_t, X_o, F, P)$.

The P - language defined by G is

$$L(G) = \left\{ (a, P(a)) \mid X_o \xrightarrow{*} a \right\}$$

where

$$P(a) = \sum_{i=1}^{N(a)} \prod_{j=1}^{K(a,i)} q_{i,j}(a)$$

where

N(a) is the number of distinct leftmost derivations of a.

K(a,i) is the number of steps in the ith derivation.

$q_{i,j}(a)$ is the probability of the rule used at the jth step in the ith derivation of a.

The value of word function P(a) is zero if 'a' is not a word in L.

II. Automata, Finite automata and Type 3 Languages, Probabilistic Automata.

II - 1 Automata

Definition II - 1 - 1

Finite Automaton is the simplest recognizer (recognizer is an another method of finitely specifying an infinite language).

A finite automaton M over an alphabet Σ is a system (K, Σ, δ, q_0, F)

where K is a finite nonempty set of states.

Σ is a finite input alphabet.

δ is a mapping of K x Σ into K.

q_0 in K is the initial state.

F $\subseteq$ K is the set of finite state.

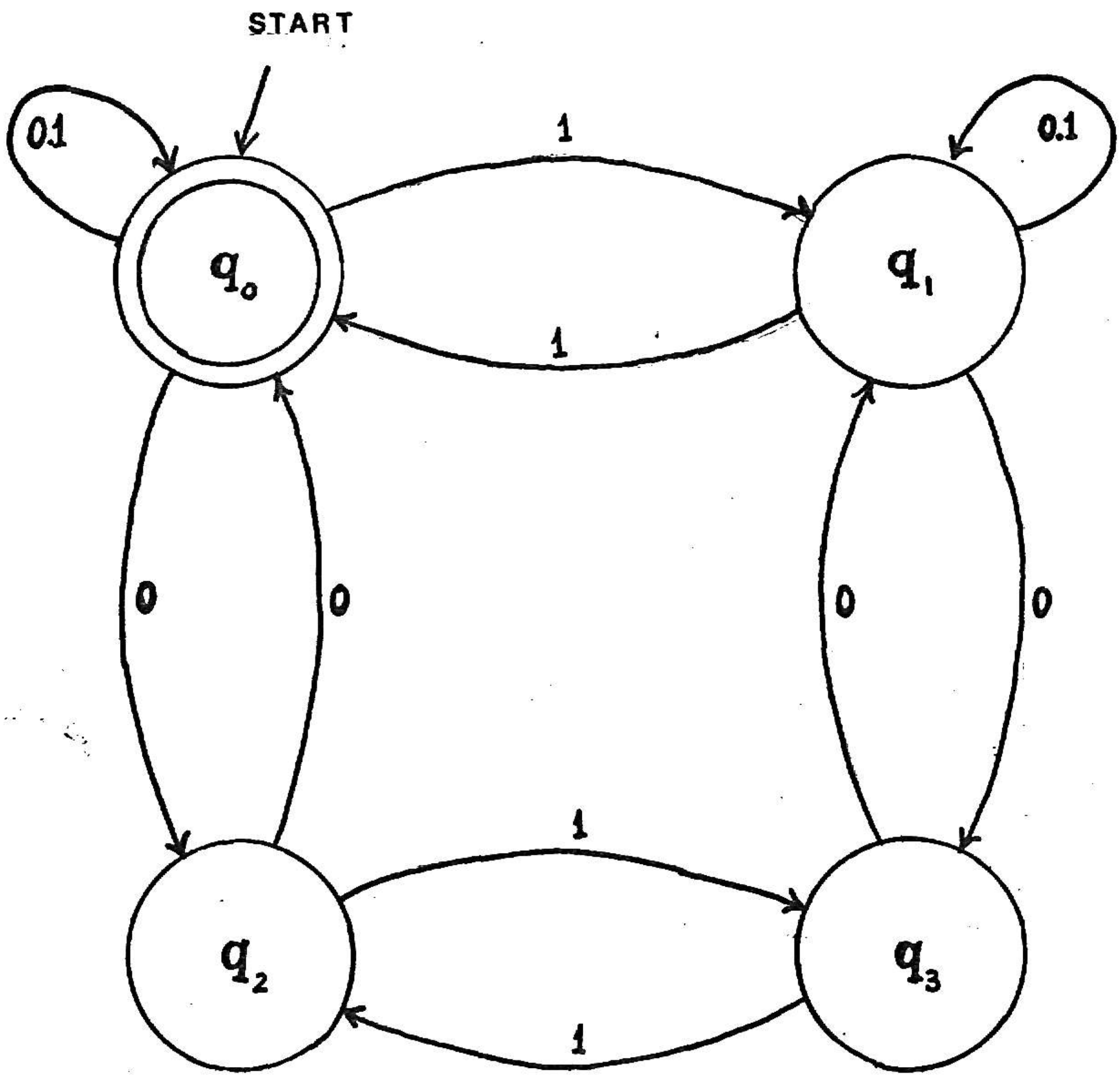
Example 2

A state diagram for the automaton is shown in Figure 1. The state diagram consists of a node for every state and a directed line from state q to state P with label a (in Σ) if the finite automaton, in state q, scanning the input symbol a. would go to state P. Final states, i.e. states in F, are indicated by a double circle. The initial state is marked by an arrow labeled start.

Definition II - 1 - 2

Nondeterministic finite automaton

M is a system (K, Σ , δ, q_0, F) where K is a finite nonempty set of states, Σ is the finite input alphabet, δ is a mapping of K x Σ into subsets of K, q_0 in K is the initial



$$M = (K, \Sigma, \delta, q_0, F)$$

$$\Sigma = \{0, 1\}$$

$$K = \{q_0, q_1, q_2, q_3\}$$

$$F = \{q_0\}$$

$$q_0 = \{q_0\} \text{ — START STATE}$$

Figure 1. State Diagram of The Finite Automaton.

state and $F \subseteq K$ is the set of final states.

The important difference between the deterministic and non-deterministic case is that $\delta_{(q,a)}$ is a (possibly empty) set of states rather than a single state. Figure 1 is nondeterministic finite automata.

II - 2 Finite Automata and Type - 3 Language

Lemma II - 2 - 1

Let $G = (V_n, V_t, X_o, F)$ be a type 3 grammar then there exists a finite automaton.

$M = (K, \Sigma, \delta, q_o, F)$ with $T(M) = L(G)$

Proof: see Reference (4)

Lemma II - 2 - 2

Given a finite automata M, there exists a type 3 grammar G such that

$L(G) = T(M)$

Proof: see Reference (4)

II - 3 Probabilistic automata

Definition II - 3 - 1

A probabilistic Finite State Automaton (stochastic finite state automaton) is a 6 tuple.

$M = (K, \Sigma, MP, q_o, \Phi, F)$

where

K, Σ, q_0 and F have the meaning as the Finite-state automaton.
$\Phi = [\phi_1, \phi_2, \cdots, \phi_n]$ is the set of all initial-state probabilities associated with the state set K.
MP is a mapping of Σ into the set of n x n probabilistic transition matrics. The interpretation of MP(a), $a \in \Sigma$, can be stated as follows. Let $MP(a) = \| P_{ij}(a) \|$ where (1) $P_{ij}(a) \geq 0$ is probability of entering state q_j from q_i under the input a. (2) $\sum_{j=1}^{n} P_{ij} = 1$ for all i=1, ..., n.

Definition II - 3 - 2

Markov Process and Transition Probabilistics

A random process J is said to be a Markov process if there exists a set of conditional probabilities $P(j_k/j_{k-r}, j_{k-r+1} \cdots j_{k-1})$. If r is the smallest integer for which this is true, the process is said to be an rth order Markov process. The conditional probabilities $P(j_k/j_{k-r} \cdots\cdots j_{k-1})$ are called transition probabilities.

If we are dealing with an rth order Markov process the probability of any sequence.

$J_k = j_1, j_2, \cdots\cdots j_r, j_{r+1} \cdots\cdots j_k$ is given by

$P(J_k) = P(j_1, j_2, \cdots\cdots j_r) . P(j_{r+1}/j_1 \cdots\cdots j_r)$
$P(j_{r+2}/j_2 \cdots\cdots j_{r+1}) \cdots\cdots\cdots\cdots P(j_k/j_{k-r} \cdots\cdots j_{k-1})$

Definition II - 3 - 3

Transient states, Absorbing states and Recurrent states.

The states of a Markov process can be classified either as transient states, absorbing states or recurrent states.

(1) Absorbing state - A state in Markov process is absorbing if once the process enter the state if never undergoes a transition to any other state.

(2) Transient state - A transient state is never entered after a certain time.

(3) Recurrent state - A recurrent state is neither absorbing nor transient.

Example 3 (see Figure 2)

Definition II - 3 - 4

Absorbing Markov Process

A Markov Process is absorbing if it contains absorbing states.

A finite-state p-grammar can be described by an absorbing Markov process.

Definition II - 3 - 5

Finite-state p-Grammar as an Absorbing Markov Process

Let $G = (V_n, V_t, X_o, F, Q)$ be a finite-state p-grammar.

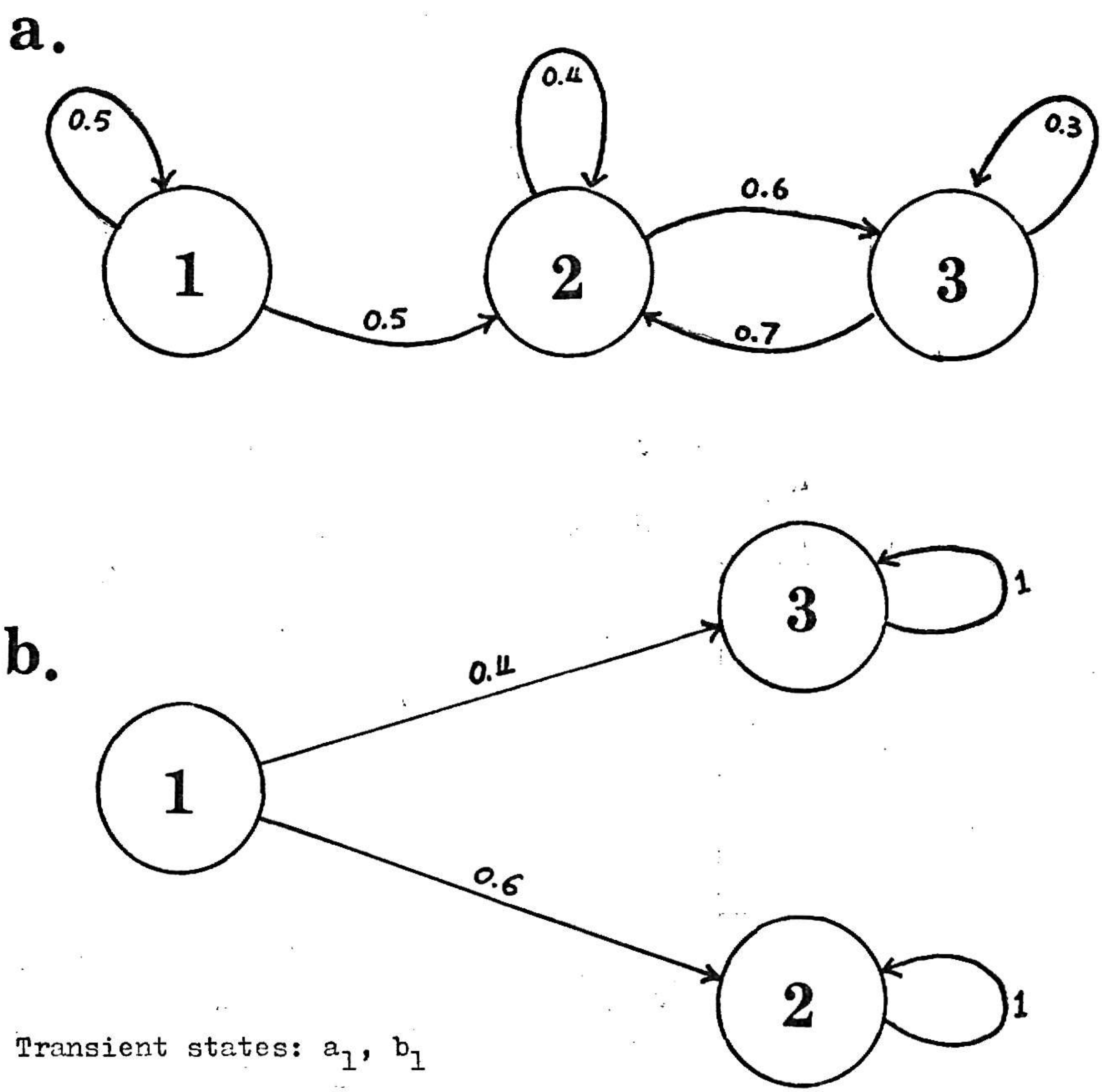
G can be represented by an absorbing Markov process with state set $V_n \cup T$ where T is a terminal state.

The transition matrix P has the following entries.

If $q: A_k \rightarrow aA_i \in Q \times F$

$A_k, A_i \in V_n, a \in V_t$

then $p_{k,i}=q$



Transient states: a_1, b_1

Absorbing states: b_2, b_3

Recurrent states: a_2, a_3

Figure 2. Diagram of States.

if $q: A_k \rightarrow a \in Q \times F \qquad A_k \in V_n, \quad a \in V_t$

then $p_{k,t} = q$

If $V_t = \{a_1, a_2, \ldots a_m\}$ the transition matrix of the process can be represented as $P = P(a_1) + P(a_2) + \ldots + P(a_m) + P(T)$

where $P(a_j)$ is the transition matrix for symbol a_j

or $p_{k,i}(a_j) = q$ if and only if $a: A_k \rightarrow a_j A_i \in Q \times F$

$$\text{and } P(T) = \begin{bmatrix} 0 & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & 0 \\ 0 & \cdot & \cdot & \cdot & 1 \end{bmatrix}$$

The current state vector $\pi(a)$ is defined as, $\pi_k(a)$ is the probability of being in state $k \in V_n \cup T$ after observing input $a \in V_t$

The initial state vector is

$\pi(\lambda) = [10..0]$

$\pi(a)$ is computed as follows

Let $a = a_1, a_2, \ldots, a_j \in V_t^+$

$\pi(a) = \pi(\lambda)\, P(a_1)\, P(a_2) \ldots P(a_j)$.

Lemma II - 3 - 1

Let G be a proper unrestricted finite-state p-grammar, G is consistent if and only if

$$\lim_{n\rightarrow\infty} P^n_{1,t} = 1$$

Proof: see Reference (6)

Lemma II - 3 - 2

If a finite-state p-grammar

$G = (V_n, V_t, X_o, F, P)$ is consistent then its corresponding Markov process has no recurrent states.

Proof: see Reference (6)

III. Parameter Estimation and Nonlinear Programming

III - 1 Parameter Estimation

In scientific investigation it frequently occurs that mathematical equations are used to describe physical phenomena, such equations may be derived from physical law or they may be convenient forms for summarizing tabulated data. In most cases the equations will contain some variables called parameters whose actual numerical values are not known. These can be determined only by analyzing physical data. One assigns to the parameters these values which make the mathematical equations best fit the physical data (5). Least squares and Maximum likelihood method are commonly employed to get these values.

One must devise a criterion of what constitutes a best fit before one under takes the estimation of the parameters.

Definition III - 1 - 1 Best Fit Conditions

Ideally one would like to find values of the θ that will satisfy equations $y_u = f(a_u, \theta)$ exactly for each experiment.

where y_u - observed variables for the uth experiment

$y_u = \{ y_{u_1}, y_{u_2}, \ldots, y_{u_k} \}$

a_u - independent variable $= \{ a_{u_1}, a_{u_2}, \ldots, a_{u_m} \}$

θ - parameters $= \{ \theta_1, \theta_2, \cdots, \theta_n \}$

Due to errors in measurement and inaccuracies in the model we can not, however, expect an exact fit. Equation $y_u = f(a_u, \theta)$ must be rewritten in the form

$$U_u = f(a_u, \theta) - y_u \qquad (1)$$

where U_u is a vector representing the departure of the predicted values $f(a_u, \theta)$ from the observed values y_u at the uth experiment. Equation (1) may also be regarded as the definition of the quantities U_u as function of the known a_u and y_u, and unknown θ. As such, the U's are called the residuals. The task of the parameter estimator is to find values of the θ which minimize (or maximize θ) some appropriate function of the U's. We denote this function $F(U_u) = F(f(a_u, \theta) - y_u)$ with the a_u and y_u given. $F(U_u)$ becomes a function of the θ's alone.

Definition III - 1 - 2 Least Squares Method

Suppose k=1 i.e. there is only one observed variable per experiment. We define

$$F_{L.S.}(U_u) = \sum_{u=1}^{n} U_u^2$$

i.e. $G_{L.S.}(\theta) = \sum_{u=1}^{n} (f(a_u, \theta) - y_u)^2$

we determine θ so as to minimize $G_{L.S.}(\theta)$. When several variables are observed at each experiment, least squares is invalid, it does not make sense to add together sums of squares of say pressure and temperatures. This problem may be overcome by assigning a weight factor to each variable and minimizing the weighted sum of squares.

$$G_{W.L.S.}(\theta) = \sum_{u=1}^{n} \sum_{i=1}^{k} W_i[f_i(a_u\theta)-y_{ui}]^2$$

or using the Maximum lkielihood method.

Definition III - 1 - 3 Maximum Likelihood

The method of maximum likelihood consists of selecting that value of the parameter θ under consideration for which $f(a_u,\theta)$, the probability (or the value of the joint density) of obtaining the sample values, is a maximum. Looking at (a_u, θ) as a value of a function of θ. We refer to it as a likelihood and to the corresponding function as the likelihood function, hence the name 'method of maximum likelihood'.

We may assume that the u's are random variables possessing a joint probability density function $P(\underset{\sim}{U}, \phi)$ of known mathematical form, possibly containin- some unknown parameter ϕ. Here u denotes the complete vector of all u's, i.e. $U = \left\{U_{11}, U_{22}, \ldots, U_{kk}\right\}$. According to the maximum likelihood principle, we seek those values of the θ and ϕ which maximize the likelihood of having made the actual observations, i.e. which maximize P (or more conveniently, its logarithm).

Thus

$G_{M.L.} = (\theta, \phi) = \log P(f(a_u, \theta) - y_u, \phi)$

If k=1 and the u's are normally distributed with zero means and covarience matrix

$E(U_u U_n) = \sigma^2 \; {}_{un} \; \left(= \begin{matrix} \sigma^2 \text{ if } u=n \\ 0 \text{ if } u \neq n \end{matrix} \right)$

where E denotes expected values and is a constant. Then it is easy to show that maximum likelihood is equivalent to least squares.

If $k > 1$ and the u's are distributed as before but with

$E(U_{ui} U_{nj}) = \frac{\sigma^2}{w_i} \delta_{un} \delta_{ij}$

the maximum likelihood reduces to weighted least squares.

III - 2 Nonlinear Programming

Maximizing the best fit criterion function may be used to solve general nonlinear programming problems.

These have the form:

Find the values of $\theta_1, \theta_2, \ldots, \theta_\ell$ that maximize a given function $G(\theta)$ subject to the constraints

$z_i(\theta) \leq 0 \qquad (i=1, 2, \ldots, r)$

The maximization algorithm used should be most efficient in cases where the maximum is at an interior point of feasible region. It suffers, however, from the disadvantages of requiring the coding of first and (sometimes) second derivatives of the objective function and that a feasible initial guess must be supplied.

III - 3 Solution of Simultaneous Nonlinear Equations

Solving the set of 1 simultaneous equations:

$g_u(\theta) = 0 \qquad (u=1, 2, \ldots, \ell)$

for the ℓ unknowns $\theta_1, \theta_2, \ldots, \theta_\ell$ is equivalent to minimizing $G(\theta) = \sum_{u=1}^{\ell} g_u^2(\theta)$

Thus, the simultaneous equation problem is equivalent to a least squares problem.

Solution Synthesis

1. Model of Meiosis (P-finite state automata)

The model of DNA (Deoxynucleic Acid) Meiosis and the sample data are taken from Reference (3). DNA Meiosis has two models. There are the symetrical Hybrid Model and the Aviemore Model respectively. In this report, only the parameters of Aviemore model were estimated (see Figure 3).

This Aviemore Model can be described by a Probabilistic Finite State Automaton and according the definition of P-finite State Automaton (Def. II-3-1). We can get the contents of this automaton

$M = (K, \Sigma , MP, q_0 , \Phi , F)$ as the following

$K = \{ I, II, III, \ldots, IX \}$ is the finite state set.

$\Sigma = \phi$ is the set of terminal symboles.

$q_0 = I$ is the initial state.

$$MP = \begin{bmatrix} Z_1 & P_1 & P_1 & P_1 & P_1 & P_2 & P_2 & 0 & 0 \\ C_3 & Z_2 & 0 & 0 & 0 & 0 & 0 & C_4 & 0 \\ C_2 & 0 & Z_3 & 0 & 0 & 0 & 0 & 0 & C_1 \\ C_1 & 0 & 0 & Z_4 & 0 & 0 & 0 & C_2 & 0 \\ C_4 & 0 & 0 & 0 & Z_5 & 0 & 0 & 0 & C_3 \\ 0 & C_2 & C_3 & 0 & 0 & Z_6 & 0 & 0 & 0 \\ 0 & 0 & 0 & C_4 & C_1 & 0 & Z_7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & Z_8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & Z_9 \end{bmatrix}$$

is the next state mapping probabilistic (since $\Sigma = \phi$ we just have only one MP)

$\Phi = \{1, 0, 0, 0, 0, 0, 0, 0, 0\}$

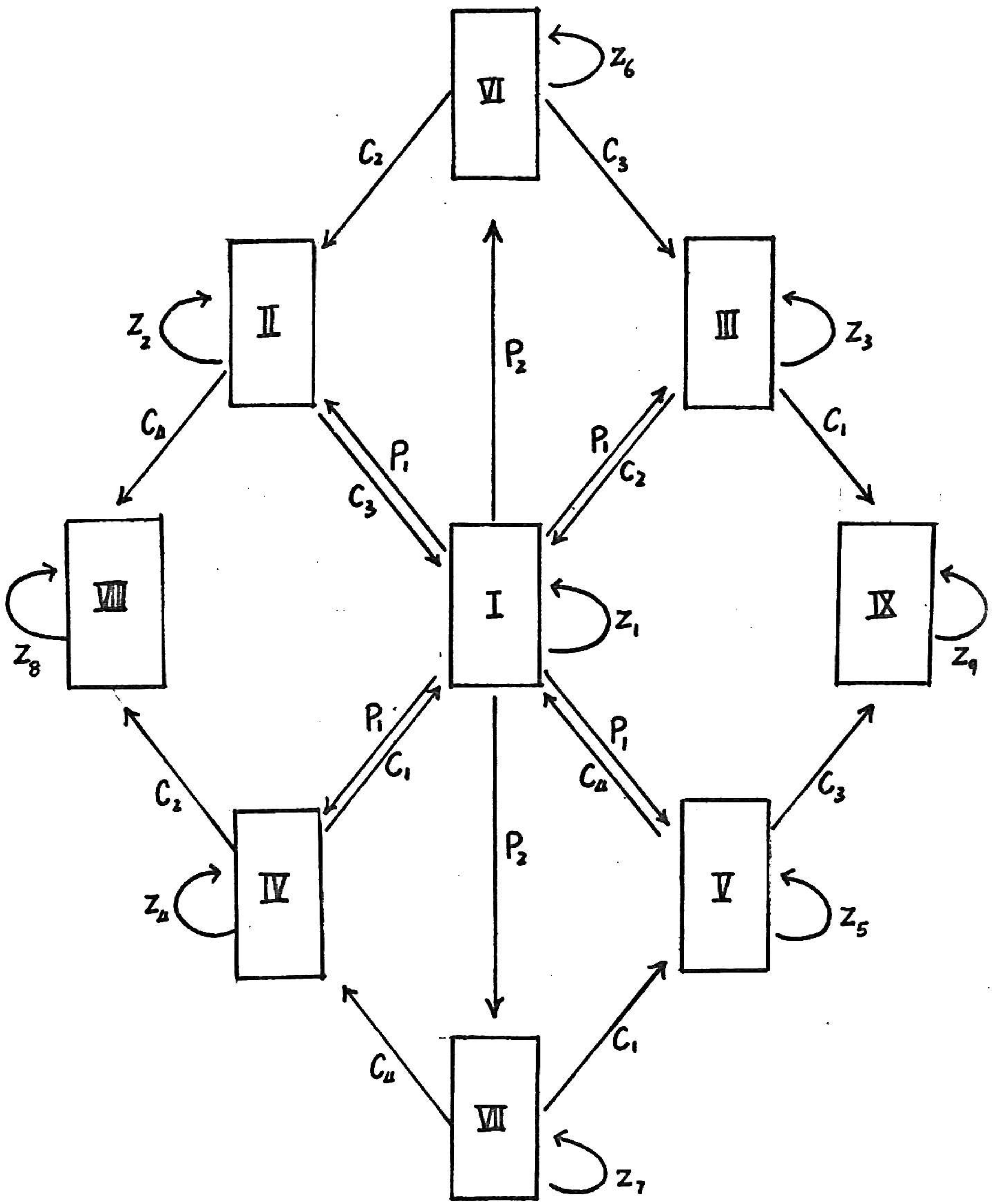


Figure 3. Aviemore Model.

$F = K$ is the set of final states.

2. Development of Equations

From the properties of P-Finite-State Automaton (Def. II-3-1), we know that $MP(a)$, $a \in \Sigma$ is defined as follows. Let $MP(a) = \| P_{ij}(a) \|$ where $P_{ij}(a) \geq 0$ is the probability of entering state q_j from state q_i under the input a, and $\sum_{i=1}^{n} Pij = 1$ for all $i=1, \ldots, n$. In this model we get the $MP(a)$ as described in the previous section. From this we obtain the following equations:

$Z_1 + 4P_1 + 2P_2 = 1$ or $Z_1 = 1 - 4P_1 - 2P_2$ 3-1

$Z_2 + C_3 + C_4 = 1$ or $Z_2 = 1 - C_3 - C_4$ 3-2

$Z_3 + C_1 + C_2 = 1$ or $Z_3 = 1 - C_1 - C_2$ 3-3

$Z_4 + C_1 + C_2 = 1$ or $Z_4 = 1 - C_1 - C_2$ 3-4

$Z_5 + C_4 + C_3 = 1$ or $Z_5 = 1 - C_3 - C_4$ 3-5

$Z_6 + C_2 + C_3 = 1$ or $Z_6 = 1 - C_2 - C_3$ 3-6

$Z_7 + C_1 + C_4 = 1$ or $Z_7 = 1 - C_1 - C_4$ 3-7

$Z_8 = 1$ 3-8

$Z_9 = 1$ 3-9

and $0 < P_1, P_2, C_1, C_2, C_3, C_4, Z_1, Z_2, \ldots, Z_9 \leq 1$

Next, we derive the probabilistic equations at final states, i.e. P(I), P(II), ..., P(IX). First we assume there are repetitive transitions from start state to some other states L times where L may be very large ($L \to \infty$) before reaching the final state. For example, if we want from I to II, first I $\rightleftarrows$ III

(L times) than I $\rightleftarrows$ V (L times) ...... and finally stop at II, so we got a variable T which represents the probability of start at I and returning to I after having traversed all possible loops from zero to an infinite number of times.

$$T = (\sum_{\ell=0}^{\infty} (P_2C_2C_3)^{\ell})^2 . (\sum_{j=0}^{\infty} (P_2C_1C_4)^{j})^2 . (\sum_{k=0}^{\infty} (P_1C_1)^k . (\sum_{m=0}^{\infty} (P_1C_2)^m)$$
$$. (\sum_{n=0}^{\infty} (P_1C_3)^n) . (\sum_{s=0}^{\infty} (P_1C_4)^s)$$

$$\because X^0 + X^1 + X^2 + \dots + X^{\infty} = \frac{1}{1-X} \quad , \quad 0 \leq X < 1$$

$$\therefore T = \frac{1}{(1-P_2C_2C_3)^2} . \frac{1}{(1-P_2C_1C_4)^2} . \frac{1}{(1-P_1C_1)} . \frac{1}{(1-P_1C_2)} . \frac{1}{(1-P_1C_3)}$$
$$. \frac{1}{(1-P_1C_4)}$$

According properties of regular grammar, we can get

$$P(I) = Z_1.T \qquad 3\text{-}10$$
$$P(II) = Z_2.T.(P_1+P_2.C_2) \qquad 3\text{-}11$$
$$P(III) = Z_3.T.(P_1+P_2.C_3) \qquad 3\text{-}12$$
$$P(IV) = Z_4.T.(P_1+P_2.C_4) \qquad 3\text{-}13$$
$$P(V) = Z_5.T.(P_1+P_2C_1) \qquad 3\text{-}14$$
$$P(VI) = Z_6.T.P_2 \qquad 3\text{-}15$$
$$P(VII) = Z_7.T.P_2 \qquad 3\text{-}16$$
$$P(VIII) = Z_8.T. (P_1C_2+P_1C_4+2P_2C_2C_4) \qquad 3\text{-}17$$
$$P(IX) = Z_9.T.(P_1C_1+P_1C_3+2P_2C_1C_3) \qquad 3\text{-}18$$

3. Experimental Data

In the given model there are only four observation variables and their relations are as follows:

$P(I) = 0.9466$ 3-19

$P(II) + P(III) + P(IV) + P(V) = 0.0079$ 3-20

$P(VI) + P(VII) = 0.0001$ 3-21

$P(VIII) + P(IX) = 0.0454.$ 3-22

Substituting 3-10 into 3-19, we get

$y_1 = T.\ Z_1 = 0.9466.$ 3-23

Substituting 3-11, 3-12, 3-13, 3-14 into 3-20, we get

$y_2 = T.(Z_2.(P_1+P_2C_2)+Z_3(P_1+P_2C_3)+Z_4(P_1+P_2C_4)+Z_5(P_1+P_2C_1)$

$= 0.0079.$ 3-24

Substituting 3-15, 3-16 into 3-21, we get

$y_3 = T.P_2(Z_6+Z_7) = 0.0001.$ 3-25

Substituting 3-17, 3-18 into 3-22, we get

$y_4 = T.(P_1C_2+P_1C_4+2P_2C_2C_4+P_1C_1+P_1C_3+2P_2C_1C_3)$

$= 0.0454.$ 3-26

4. Selection of Estimation Technique

There are six unknown variables P_1, P_2, C_1, C_2, C_3, C_4 but only four nonlinear equations 3-23, 3-24, 3-25, 3-26. Therefore an infinity of solutions may exist in this case. We can not solve these equations directly, but try to estimate the values of these unknown variables to approximate the observed variable as closely as possible to the given data provided in DNA.

Due to the fact of independent variables and insufficient experimental data, neither Maximum likelihood estimate method nor Least squares method can be used. Although attempts were made to estimate these parameters by using the given observed data several times, no answer could be obtained. Then, the nonlinear programming method was considered. Because there are four observed objective functions, we can use solution of simultaneous nonlinear equations method. So, first, the four equations 3-23, 3-24, 3-25, 3-26 would be changed to the following:

$$y_1' = y_1 - 0.9466 \qquad 3\text{-}27$$

$$y_2' = y_2 - 0.0079 \qquad 3\text{-}28$$

$$y_3' = y_3 - 0.0001 \qquad 3\text{-}29$$

$$y_4' = y_4 - 0.0454 \qquad 3\text{-}30$$

Second, summing the squares of the above four equations, and then minimizing, we obtain:

$$G = ((y_1')^2+(y_2')^2+(y_3')^2+(y_4')^2) \times (-1) \qquad 3\text{-}31$$

We assume equation G, 3-31 to be an objective function, and let 3-1, 3-2, ..., 3-9 and $0 < P_1, P_2, C_2, C_3, C_4, Z_1, Z_2, \ldots, Z_9 \leq 1$ be the constraint functions. We can choose any initial guesses which are within these constraint conditions, substitute them into 3-31 and modify the initial guesses to maximize the equation until it approaches zero.

In solving this problem, the Nonlinear Parameter Estimation and Programming program (5) can be used. Details of this program are presented in Appendix A.

Results

1. Estimates from Nonlinear Program

Several runs of the non-linear program were made with various initial guesses, see Table 1.

Using P_1=0.1, P_2= 0.025, C_1=0.05, C_2=0.05, C_3=0.05, C_4=0.05 as initial parameter guesses (input data) to run the non-linear programming program (Appendix A) the parameters were obtained as following:

P_1 = 0.02342779

P_2 = 0.0000001506

C_1 = 0.4389197

C_2 = 0.4809957

C_3 = 0.4809954

C_4 = 0.4389207

It is the best set of parameters among the sets of parameters in Table 1, because its max. of objective function is largest, -0.1938959E-6, and very close to zero.

For checking whether these results approximated the observed data, they were substituted into 3-1, 3-2, ..., 3-18 and the following answers were obtained

P(I) = 0.9464277   4-1

P(II) + P(III) + P(IV) + P(V) = 0.007837183   4-2

P(VI) + P(VII) = 0.00000002518968   4-3

P(VIII) + P(IX) = 0.04501232   4-4

Comparing these equations with 3-19, 3-20, 3-21, 3-22

Table 1

No.	Initial parameter guesses	Parameter	Maximum of objective function	(1)	(2)	(3)
1	$P_1=0.1$	.02491752	$-.5467875\text{x}10^{-4}$	113	19	1.52
	$P_2=0.1$	$.1313012\text{x}10^{-8}$				
	$C_1=0.1$	.3937789				
	$C_2=0.1$	.4854715				
	$C_3=0.1$	.4854715				
	$C_4=0.1$	.3937789				
2	$P_1=0.01$	.01947914	$-.7493161\text{x}10^{-2}$	152	20	1.66
	$P_2=0.005$	.04882335				
	$C_1=0.05$	.2233725				
	$C_2=0.05$	.4999937				
	$C_3=0.05$	.4999997				
	$C_4=0.05$	.2233723				
3	$P_1=0.01$	.03215938	$-.2442538\text{x}10^{-2}$	2191	258	17.63
	$P_2=0.001$	$.9294912\text{x}10^{-2}$	$-.2442538\text{x}10^{-2}$			
	$C_1=0.1$	.3145316				
	$C_2=0.1$	.5000025				
	$C_3=0.1$	.4999973				
	$C_4=0.1$	.3145357				
4	$P_1=0.01$	.02277552	$-.2965198\text{x}10^{-3}$	194	31	1.14
	$P_2=0.005$	$.1421340\text{x}10^{-6}$				
	$C_1=0.45$	.3268077				
	$C_2=0.45$	.4438046				
	$C_3=0.45$	.4438068				
	$C_4=0.45$	.3268085				

No.	Initial parameter guesses	Parameter	Maximum of objective function	(1)	(2)	(3)
5	P_1=0.001	.008684143	$-.9601931\times10^{-3}$	159	21	1.96
	P_2=0.001	.01554924				
	C_1=0.005	.1957616				
	C_2=0.005	.499999				
	C_3=0.005	.4999996				
	C_4=0.005	.1957619				
6	P_1=0.1	.05792904	$-.2409731\times10^{1}$	154	23	1.94
	P_2=0.01	.001865075				
	C_1=0.025	.4711328				
	C_2=0.025	.5047635				
	C_3=0.025	.4952363				
	C_4=0.025	.4708962				
7	P_1=0.1	.02340817	$-.2464492\times10^{-6}$	203	30	2.17
	P_2=0.05	$.77248\times10^{-7}$				
	C_1=0.05	.4390432				
	C_2=0.05	.4802026				
	C_3=0.05	.4802024				
	C_4=0.05	.4390437				
8	P_1=0.1	.02342779	$-.1938959\times10^{-6}$	177	29	2.17
	P_2=0.025	$.1506\times10^{-6}$				
	C_1=0.05	.4389197				
	C_2=0.05	.4809957				
	C_3=0.05	.4809954				
	C_4=0.05	.4389207				

Note: (1) - Function evaluation times
(2) - Derivation evaluation times
(3) - Running time (sec)

and the sum of weighted squared error for these values and the experimental data is as following:

$$\mathrm{ERR} = \frac{(0.9464277-0.9466)^2}{0.9466} + \frac{(0.007837183-0.0079)^2}{0.0079}$$

$$+ \frac{(0.0000002518968-0.0001)^2}{0.0001} + \frac{(0.04501232-0.0454)^2}{0.0454}$$

$$= 0.0000141126$$

It showed that the four values were very close respectively which implied that parameters were good enough.

2. Sensitivity to Initial Guesses

From Table 1, we can see that the estimated parameters, function evaluation and derivation evaluation depend on the initial parameter guesses. For example, in No. 3 of Table 1, the initial parameter guesses P_1=0.01, P_2=0.001, C_1=C_2=C_3=C_4 =0.1, function evaluation occurs 2191 times. Derivative evaluation occurs 258 times. The maximum of objective function is -0.2442538E-2. This implies that these initial parameter guesses were not good. But in No. 8 the initial parameter guesses P_1=0.1, P_2=0.025, C_1=C_2=C_3=C_4=0.05, function evaluation occurs only 177 times. The derivative evaluation occurs only 29 times. The maximum of objective function is -0.1938959E-6. This indicates that these initial parameters were good.

In short each different initial guess results in a different set of estimates that may or may not be acceptable.

In order to obtain a satisfactory answer, the choice of initial guesses is very important. Unfortunately, except that the initial guesses be within the condition range, there is not any general rule for choosing initial guesses. We have to try one by one until the satisfactory parameters are obtained.

3. Normalization

From 4-1, 4-2, 4-3, 4-4, we can see that the obtained observed variables are close to the given values. However, the obtained observed variable of 4-3 is too small, i.e. the given value of this observed variable is 0.0001, but the obtained value is 0.00000001. It means that the obtained parameter can be modified. Instead of choosing other initial guesses to get the better one, we consider modifying the objective function, 3-33. We found in 3-33 the y_3' is too small, i.e. $y_3' = y_3 - 0.0001$. Because of this fact the non-linear programming program just modified the other terms of the objective function to maximize the objective function. Therefore to solve this problem, normalization of these equations is required. Normalization entails dividing the equations y_1', y_2', y_3', y_4' by a proportionality coefficient. So we modify the equations 3-23, 3-24, 3-25, 3-26 and get the following:

$$y_1'' = \frac{y_1}{0.9466} = 1 \quad \text{or} \quad y_1'' - 1 = 0 \qquad \text{4-5}$$

$$y_2'' = \frac{y_2}{0.0079} = 1 \quad \text{or} \quad y_2'' - 1 = 0 \qquad \text{4-6}$$

$$y_3'' = \frac{y_3}{0.0001} = 1 \quad \text{or} \quad y_3'' - 2 = 0 \qquad \text{4-7}$$

$$y''_4 = \frac{y_4}{0.0454} = 1 \quad \text{or} \quad y''_4 - 1 = 0 \qquad 4\text{-}8$$

and

$$y' = ((y''_1-1)^2+(y''_2-1)^2+(y''_3-1)^2+(y''_4-1)^2.(-1) \qquad 4\text{-}9$$

Equation 4-9 is a new objective function, we can write a new subroutine in nonlinear programming program (Appendix A). Table 2 is the result of new objective function. No. 7 in Table 2 is the best (the maximum of objective function is largest and very close to zero, i.e. 0.6106013E-06). The obtained parameters are following:

P_1 = 0.02334443

P_2 = 0.000597968

C_1 = 0.4735852

C_2 = 0.4462720

C_3 = 0.4462446

C_4 = 0.4736139

To check the estimated parameters we put them into 3-1, 3-2, ..., 3-18 and get the following

P(I) = 0.945862 4-9

P(II)+P(III)+P(IV)+P(V) = 0.007899995 4-10

P(VI)+P(VII) = 0.0001000002 4-11

P(VIII)+P(IX) = 0.0453973 4-12

And the sum of weighted error for these values and the experimental data is following

ERR = 0.0000005754

Table 2

No.	Initial parameter guesses	Paramater	Maximum of objective function	(1)	(2)	(3)
1	$P_1=0.2$	.00249214	-.9604318	238	34	2.58
	$P_2=0.05$	$.6197309\times10^{-4}$				
	$C_1=0.05$	.09679675				
	$C_2=0.05$	.09678811				
	$C_3=0.05$	.09679353				
	$C_4=0.05$	.09680402				
2	$P_1=0.05$	.002534017	-.9570221	330	42	3.82
	$P_2=0.05$	$.6289206\times10^{-4}$				
	$C_1=0.05$	.1030563				
	$C_2=0.05$	.1030629				
	$C_3=0.05$	.1030629				
	$C_4=0.05$	.1030629				
3	$P_1=0.1$	.02915805	-.2109516	213	31	2.27
	$P_2=0.01$	$.429047\times10^{-3}$				
	$C_1=0.001$	.3999013				
	$C_2=0.001$	.5335436				
	$C_3=0.001$	.4664556				
	$C_4=0.001$	.4371563				
4	$P_1=0.1$	.0212735	$-.7444972\times10^{-1}$	165	24	1.82
	$P_2=0.01$	$.3896547\times10^{-3}$				
	$C_1=0.05$	.4072636				
	$C_2=0.05$	.4881583				
	$C_3=0.05$	.5118416				
	$C_4=0.05$	.406245				

No.	Initial parameter guesses	Parameter	Maximum of objective function	(1)	(2)	(3)
5	$P_1=0.1$	.01719518	-.1363227	115	18	1.41
	$P_2=0.0025$	$.3288405\times10^{-3}$				
	$C_1=0.05$	.3770843				
	$C_2=0.05$	.4917224				
	$C_3=0.05$	.5082775				
	$C_4=0.05$	.3770097				
6	$P_1=0.1$	.02915805	-.2109516	213	31	2.5
	$P_2=0.01$	$.429047\times10^{-3}$				
	$C_1=0.001$	.3999013				
	$C_2=0.001$	.5335439				
	$C_3=0.001$	.4664556				
	$C_4=0.001$	.4371563				
7	$P_1=0.01$	.02334443	$-.6106013\times10^{-6}$	297	45	3.2
	$P_2=0.001$	$.597968\times10^{-3}$				
	$C_1=0.1$	.4735852				
	$C_2=0.1$	.446272				
	$C_3=0.1$	.4464446				
	$C_4=0.1$	.4736139				
8	$P_1=0.001$	.02334388	$-.6112633\times10^{-6}$	310	43	3.23
	$P_2=0.001$	$.5979487\times10^{-3}$				
	$C_1=0.005$	.457405				
	$C_2=0.005$	.4625061				
	$C_3=0.005$	.4625273				
	$C_4=0.005$	.4574719				

Note: (1) - Function evaluation times
(2) - Derivation evaluation times
(3) - Running time (sec)

These obtained observed variable are very close to the data. This means that the parameters which were obtained by normalization equation are more satisfactory than those previously obtained. It can be seen that normalization improves the accuracy of the estimation procedure.

4. Simplification

Equation 3-23, 3-24, 3-25, 3-26 were developed under the condition of assuming that there are repetitive transitions from start state to some state before the transition from start state to the final state. But this assumption greatly complicated derivation from this model. For this reason, we try to simplify these equations. We obtain the new y_1, y_2, y_3, y_4 equations as following:

$$y_1 = (1-4P_1-2P_2)(2P_2\cdot(C_2C_3+C_1C_4) + P_1(C_1+C_2+C_3+C_4)+1) \qquad 4\text{-}13$$

$$y_2 = (1-C_3-C_4)(P_1+P_2C_2)+(1-C_1-C_2)(P_1+P_2C_3)+(1-C_1-C_2)(P_1+P_2C_4) + (1-C_3-C_4)(P_1+P_2C_1) \qquad 4\text{-}14$$

$$y_3 = 2P_2 \qquad 4\text{-}15$$

$$y_4 = P_1C_2+P_1C_4+2P_2C_2C_4+P_1C_1+P_1C_3+2P_2C_1C_3 \qquad 4\text{-}16$$

We can use these four equations to estimate the parameters as in the previous situations. Table 3 contains the results of these estimation runs.

In Table 3 the best run is No. 5, the results are the following:

$P_1 = 0.02438452$

Table 3

No.	Initial parameter guesses	Parameter	Maximum of objective function	(1)	(2)	(3)
1	$P_1=0.001$	.0248853	$-.8729748\times10^{-2}$	804	125	8.62
	$P_2=0.001$	$.5774652\times10^{-3}$				
	$C_1=0.005$	.414717				
	$C_2=0.005$	.4866133				
	$C_3=0.005$	.5133867				
	$C_4=0.005$	.4138312				
2	$P_1=0.1$	.02404754	$-.7823654\times10^{-2}$	195	29	2.21
	$P_2=0.0025$	$.5440307\times10^{-3}$				
	$C_1=0.05$	.7934365				
	$C_2=0.05$	.03622				
	$C_3=0.05$	.9637799				
	$C_4=0.05$	.03547687				
3	$P_1=0.1$	.02385641	-.5428308	673	97	6.09
	$P_2=0.001$	$.1607722\times10^{-3}$				
	$C_1=0.05$	.4280294				
	$C_2=0.05$	.347324				
	$C_3=0.05$	.5639734				
	$C_4=0.05$	.4360262				
4	$P_1=0.2$	.002515124	-.9587677	173	24	1.9
	$P_2=0.05$	$.6247922\times10^{-4}$				
	$C_1=0.05$	.09987652				
	$C_2=0.05$	.09987062				
	$C_3=0.05$	.09987646				
	$C_4=0.05$	.09987652				

No.	Initial parameter guesses	Parameter	Maximum of objective function	(1)	(2)	(3)
5	P_1=0.1	.02438452	$-.22416152\times10^{-4}$	915	129	8.73
	P_2=0.05	$.6247372\times10^{-3}$				
	C_1=0.05	.4953973				
	C_2=0.05	.417259				
	C_3=0.05	.431041				
	C_4=0.05	.496227				
6	P_1=0.1	.01618117	-.3267009	218	34	2.62
	P_2=0.01	$.875415\times10^{-3}$				
	C_1=0.05	.436805				
	C_2=0.05	.4920012				
	C_3=0.05	.5079966				
	C_4=0.05	.4339736				
7	P_1=0.1	.003446633	-.8875548	216	29	2.23
	P_2=0.1	$.8298822\times10^{-4}$				
	C_1=0.1	.1987585				
	C_2=0.1	.1987685				
	C_3=0.1	.1987554				
	C_4=0.1	.7987586				
8	P_1=0.1	.00251563	-.958729	144	20	1.69
	P_2=0.1	$.6249164\times10^{-4}$				
	C_1=0.05	.0999462				
	C_2=0.05	.09994566				
	C_3=0.05	.09994626				
	C_4=0.05	.0999462				

Note: (1) - Function evaluation times
(2) - Derivation evaluation times
(3) - Running time (sec)

$P_2 = 0.0006247372$

$C_1 = 0.4953973$

$C_2 = 0.417259$

$C_3 = 0.431041$

$C_4 = 0.496227$

The computation of the observed variables yields

$P(I) = 0.9421246$

$P(II)+P(III)+P(IV)+P(V) = 0.0078987$

$P(VI)+P(VII) = 0.0001000051$

$P(VIII)+P(IX) = 0.04539117$

and the sum of weighted squared error for these values and the experimental data is

ERR = 0.000021161

comparing these estimated parameters with the parameters estimated by the normalization equations (in section IV-3) we find that they were approximately the same. Therefore the simplified model produces estimates that are very similar to those of the full model with less computation.

5. Comparison of Running Times

Through Table 1 to Table 3, we can find the running time of parameter estimation depends on the times of function evaluation and derivation evaluation, such as, No. 3 of Table 1, it took 17.63 seconds, and No. 7 of Table 1, it took 1.14 second. The difference is 16.49 seconds. We also can find No. 8 of Table 1 which is the best result in Table 1 took 2.17 second, No. 7

of Table 2 which is the best result in TAble 2 took 3.20 seconds. And No. 5 of Table 3 took 8.73 seconds.

It means that the running time of execution does not concern with the method of parameter estimation. The normalization and simplification can not decrease the executing time, but only modify the result or simply the equation.

Conclusion

In this report, properties and rules of both probabilistic finite state automata and probabilistic regular grammars were employed to develop the equations of a given probabilistic finite state automaton model (Model of DNA). A nonlinear programming technique was used to estimate parameters of these equations from experimental data. The parameters which were estimated were probabilities of the state transaction of the given model.

From the results, it can be seen that the method which was discussed in this paper can be used to estimate parameters of any probabilistic finite state automata model. Therefore, the other model - symmetrical Hybrid Model, which was not discussed in this report, can also be estimated in a similar manner. That is, first develop the equations of parameters which will be estimatea, and then estimate the parameters. The simplication method can be used to formulate the equations rather than assuming the model has repetitive transitions. The equations must then be normalized. After obtaining the objective function; the nonlinear programming technique can be used to estimate the parameters directly. We have learned in this research that the Maximum likelihood and Least squares method can not be used. We also have learned that an optimum feasible initial guess is necessary to reach the accurate estimation. The author believes that satisfactory parameters can be obtained with a reasonable number of guesses.

One shortage of this method is that the different estimated value would be obtained from the different initial guesses, therefore we have to try various initial guesses until a satisfactory value is reached. It may be a lengthy but not difficult job.

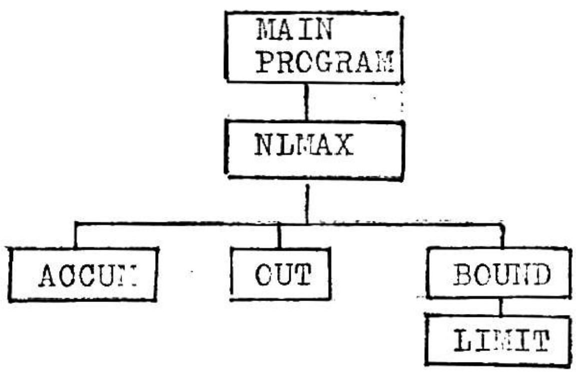
The author mastered the basic concents of Finite-state automata, and learned several techniques of parameter estimation such as maxlikelihood, least square and nonlinear programming during the course of this research.

Appendix A

Nonlinear Programming Program

The program is a subset of the Nonlinear Parameter Estimation and Programming (5). Only those portions that are used in this program are explained.

1. Structure of the program



MAIN PROGRAM: Calls on subroutine NLMAX.

NLMAX: Reads general input. Finds and prints out maximum of objective function.
calls on ACCUM, OUT.
This program uses the Davidon-Fletcher-Newell method.

ACCUM: Computes value of objective function and its derivatives. User must write his or her own subroutine for arbitrary nonlinear programming problems.

OUT: Provides additional detailed output after soluion has been found.
In this program, it is a dummy subroutine.

BOUND: Computes the constraint penalty functions and their derivatives.
Calls LIMIT (for arbitrary constraints)

LIMIT: Computes constraint functions and their first derivatives, written by user.

2. List of synbols - common storage

Segment A

COMMON C(u,u),G1(u,u),PSCA,G(ℓ,ℓ),F(ℓ),Y(ℓ),DGV(ℓ),FF(ℓ),
TITLE(20),CUB(v),PLN(ℓ),NCON,LOUT,F3,NTH,F6,F7,METH,MD,LS,
C1(ℓ)

Segment B

COMMON V(k,k),QY(k),YTH(k,ℓ),A(n,k+m),ICOV,DET,IDER,M,NY,NA

k = number of observed variables

ℓ = number of unknown parameters

m = number of independent variables

n = number of experiments

t = number of constraints

TITLE(20)	80 characters of BCD identification
NCON	t (number of constraints)
LOUT	$=\begin{Bmatrix}1\\2\end{Bmatrix}$ if $\begin{Bmatrix}\text{iteration by iteration}\\\text{final only}\end{Bmatrix}$ output is desired
F3	G (the objective function)
NTH	ℓ (number of unknown parameters)
MD	model number
C1(ℓ)	θ_i unknown parameter

v = max(1, t/2) (i.e. 2v must be at least t)

YTH(k,ℓ) $\partial Y/\partial \theta$

3. User written subroutines

(1) SUBROUTINE ACCUM(II)

Requires: COMMON segment A

Purpose: To complete the objective function to be maximized $G(\theta_1, \theta_2, \ldots\ldots)$ and its first derivatives. The value of θ_I is found in location C1(I).

If II = 1 : place G in location F3

If II = 2 : place G in location F3

(2) SUBROUTINE LIMIT (II)

Requires: COMMON segment A

Purpose: To compute the Ith constraint function $Z_I(\theta_1, \theta_2, \ldots\ldots)$ and its first derivatives. The constraint functions must be defined so as to be negative in the interior of the feasible region. The value of θ_J is found in location C1(J).

If II = 1 : place Z_I in location X

If II = 2 : place Z_I in location X

place $\dfrac{\partial Z_I}{\partial \theta_J}$ in location CUB(J),

for J = i, 2, ... NTH

4. Input data

(1) NLMAX

Title card: Any 80 columns of BCD information, to appear as title for the output.

Format: 20A4

Problem definition integers

Containing three integers

NTH = number of unknown parameters

LOUT = 1 if iteration by iteration output is desired

2 if only final output is desired

MD = model number

Format: 3I5

Parameter initial guesses card(s)

Each containing up to eight fixed or floating point numbers.

$C^1(I)$ I = 1, 2, ... Nth initial guesses for the unknown parameter

Format: 8E10.5

(2) BOUND

NCON = number of constraints

Format: I5

5. Output

NLMAX

(1) Initial output

Title and model number

(2) Intermediate output

(omitted when LOUT = 2)

For each iteration and function evaluation: The value of the objective function and current values of the parameters.

(3) Final output

Maximum value of objective function, and the corresponding values of the parameter number of function and derivative evaluations required; the gradient of the objective function and its Hessian.

BOUND

(1) Initial output

The coefficients of the constraints penalty functions (the r_i of equation)

(2) Final output

The values of the constraints for final values of the parameters.

```
C DECK 01
C MAIN PROGRAM
 1     WRITE(6,2)
 2     FORMAT(1H1)
       CALL NLMAX
       END
```

```
C DECK 04
C COMPUTES SCALED EIGENVALUES AND VECTORS
      SUBROUTINE EIG(N,II)
      COMMON A(20,20),V(20,20),PSCA
      DIMENSION SCA(20)
      PSCA=0.
      IF(N-1)107,107,108
  107 GO TO(103,109),II
  109 V(1,1)=1.
      RETURN
  108 VN=2.
      SUM1=0.
      DO 22 I=1,N
      IF(A(I,I))101,100,101
  100 SCA(I)=1.
      GO TO 22
  101 A1=ABS(A(I,I))
      SCA(I)=1./SQRT(A1)
      PSCA=PSCA+ALOG(A1)
      DO 102 J=1,N
      A(I,J)=A(I,J)*SCA(I)
  102 A(J,I)=A(I,J)
      A(I,I)=A(I,I)*SCA(I)
   22 CONTINUE
      DO 1 I=2,N
      DO 1 J=2,I
    1 SUM1=SUM1+A(I,J-1)*A(I,J-1)
      SUM1=SQRT(2.*SUM1)
      SUME=SUM1/10.E7
      GO TO(30,31),II
   31 DO 32 I=1,N
      DO 50 J=1,N
   50 V(I,J)=0.
   32 V(I,I)=1.
   30 IN=0
      IF(N-1)18,17,18
   18 SUM1=AMAX1(SUME,SUM1/VN)
   16  CONTINUE
      DO 3 J=2,N
      J1=J-1
      DO 3 I=1,J1
      IF(ABS(A(I,J))-SUM1)3,3,4
    4 IN=1
      Y1=-A(I,J)
      Y2=(A(I,I)-A(J,J))/2.
      OMEGA=Y1/SQRT(Y1**2+Y2**2)
      OMEGA=OMEGA*SIGN(1.,Y2)
      Y1=OMEGA/SQRT(2.+2.*SQRT(1.-OMEGA**2))
      BB1=Y1**2
      BB2=1.-BB1
      Y2=SQRT(BB2)
    9 DO 5 K=1,N
      IF (K-I) 6,5,6
    6 IF (K-J) 7,5,7
    7 Y3=A(K,I)*Y2-A(K,J)*Y1
      A(K,J)=A(K,I)*Y1+A(K,J)*Y2
      A(K,I)=Y3
      A(J,K)=A(K,J)
      A(I,K)=A(K,I)
    5 CONTINUE
```

```
      BB3=2.*Y1*Y2*A(I,J)
      Y3=A(I,I)*BB2+A(J,J)*BB1-BB3
      Y4=A(I,I)*BB1+A(J,J)*BB2+BB3
      A(I,J)=(A(I,I)-A(J,J))*Y1*Y2+A(I,J)*(BB2-BB1)
      A(J,I)=A(I,J)
      A(I,I)=Y3
      A(J,J)=Y4
      GO TO(3,20),II
   20 DO 12 K=1,N
      Y3=V(K,I)*Y2-V(K,J)*Y1
      V(K,J)=V(K,I)*Y1+V(K,J)*Y2
   12 V(K,I)=Y3
    3 CONTINUE
      IF (N-2) 17,17,21
   21 IF (IN-1) 14,15,15
   15 IN=0
      GO TO 16
   14 IF (SUM1-SUME)17,17,18
17    GO TO(103,104),II
104   DO 105 I=1,N
      DO 105 J=1,N
105   V(I,J)=V(I,J)*SCA(I)
103   RETURN
      END
```

```
C DECK C8
C ARBITRARY CONSTRAINTS ON PARAMETERS
      SUBROUTINE BOUND(II,H)
      COMMON C(20,20),G1(20,20),PSCA,G(20,20),F(20),Y(20),EGV(20),FF(20)
     1,TITLE(20),CUB(20),CLB(20),PNL(20),NCON,LOUT,F3,NTH,F6,F7,METH,NPH
     2,MD,LS,C1(20)
      GO TO(1,1,2,3,23,1001,28),II
 1001 DO 1002 I=1,NCON
 1002 CLB(I)=.1*CLB(I)
      RETURN
 23   H1=H
      H2=2.*H1
      J=1
      DO 16 K=1,4
      H3=(H1+H2)*.5
      DO 17 I=1,NTH
 17   C1(I)=FF(I)+Y(I)*H3
      DO 18 I=1,NCON
      CALL LIMIT(1,J,X)
      IF(X)19,20,20
 19   IF(J-NCON)21,22,22
 22   J=1
      GO TO 18
 21   J=J+1
 18   CONTINUE
      H1=H3
      GO TO 16
 20   H2=H3
 16   CONTINUE
      H=H1
      RETURN
 1    DO 4 I=1,NCON
      CALL LIMIT(II,I,X)
      X1=CLB(I)/X
      F3=F3+X1
      GO TO(4,5),II
 5    X1=X1/X
      X=2./X*X1
      DO 6 J=1,NTH
      F(J)=F(J)-X1*CUB(J)
C THE FOLLOWING STATEMENTS MAY BE REMOVED FOR DAVIDON'S METHOD
      GO TO(100,6),METH
 100  DO 101 K=J,NTH
 101  G(J,K)=G(J,K)+X*CUB(J)*CUB(K)
C END OF REMOVABLE STATEMENTS
 6    CONTINUE
 4    CONTINUE
      RETURN
 3    J=1
 13   DO 7 I=1,NTH
 7    C1(I)=FF(I)+Y(I)*H
      DO 8 I=1,NCON
      CALL LIMIT(1,J,X)
      IF(X)10,9,9
 10   IF(J-NCON)11,12,12
 12   J=1
      GO TO 8
 11   J=J+1
 8    CONTINUE
      DO 15 I=1,NTH
```

```
15    C1(I)=FF(I)
      RETURN
9     H=.5*H
      GO TO 13
2     READ(5,200)NCON
200   FORMAT(16I5)
      CALL LIMIT(3,I,X)
      DO 50  I=1,NTH
50    FF(I)=C1(I)
      DO 14  I=1,NCON
64    CALL LIMIT(2,I,X)
      IF(X)25,55,26
55    DO 60  J=1,NTH
60    C1(J)=C1(J)-.0001*CUB(J)
      DO 61  J=1,NCON
      CALL LIMIT(1,J,H)
      IF(H)61,26,26
61    CONTINUE
      DO 63  J=1,NTH
63    FF(J)=C1(J)
      GO TO 64
25    H=0.
      DO 51  J=1,NTH
51    H=H+CUB(J)*CUB(J)
      H=X/H
      DO 53  K=1,6
      DO 52  J=1,NTH
      CUB(J)=H*CUB(J)
52    C1(J)=FF(J)+CUB(J)
      DO 54  J=1,NCON
      CALL LIMIT(1,J,X1)
      IF(J-I)56,57,56
57    X=AMIN1(X,X1)
56    IF(X1)54,59,59
54    CONTINUE
53    H=2.
59    CLB(I)=.01*(.001+ABS(X))
      DO 65  J=1,NTH
65    C1(J)=FF(J)
14    CONTINUE
      WRITE(6,70)(I,CLB(I),I=1,NCON)
70    FORMAT(31H0CONSTRAINT PENALTY COEFFICIENT/(I11,E20.6))
      RETURN
28    WRITE(6,29)
29    FORMAT(27H0CONSTRAINT              VALUE/)
      DO 30  I=1,NCON
      CALL LIMIT(1,I,X)
30    WRITE(6,31)I,X
31    FORMAT(I11,E16.6)
      RETURN
26    WRITE(6,27)I,X
27    FORMAT(35H0INITIAL GUESSES VIOLATE CONSTRAINTI5,8H   VALUEE16.6/3
     2H ******************************** )
      LS=3
      RETURN
      END
```

```
C DECK 09
C MAXIMIZES FUNCTION USING DAVIDON-FLETCHER-POWELL METHOD
      SUBROUTINE NLMAX
      COMMON C(20,20),G1(20,20),PSCA,C(20,20),F(20),Y(20),EGV(20),FF(20)
     1,TITLE(20),CUB(20),CLB(20),PNL(20),NCON,LOUT,F3,NTH,F6,F7,METH,NPH
     2,MD,LS,C1(20)
      EQUIVALENCE(NTH,L)
      METH=2
      EPSIL=.003
      KKK=1
      READ(5,2000)TITLE
 2000 FORMAT(20A4)
      READ(5,2002)NTH,LOUT,MD
 2002 FORMAT(16I5)
      WRITE(6,2001)TITLE,MD
 2001 FORMAT(1H120A4/6H0MODELI5)
      READ(5,2003)(C1(I),I=1,NTH)
 2003 FORMAT(8E10.5)
      LS=1
      CALL ACCUM(3)
      CALL BOUND(3,H)
      WRITE(6,5000)(C1(I),I=1,L)
 5000 FORMAT(26H0PARAMETER INITIAL GUESSES/(7E16.6))
      IF(LS-3)199,907,199
 199  IPH=2
      NF=0
      ND=0
      EPS=1.E-4
      EPS1=1.E-3
      AL=1.E-5
      BL=.01**L
      DO 906 I=1,L
      FF(I)=C1(I)
 906  Y(I)=-C1(I)
      H=1.
      CALL BOUND(4,H)
      DO 911 I=1,L
 911  Y(I)=C1(I)*H
      NPH=1
      IF(NCON)1,899,1
 899  NPH=2
      GO TO 16
 1    GO TO(212,16),LOUT
 212  WRITE(6,1001)
 1001 FORMAT(26H1PENALTY FUNCTION INCLUDED)
 16   INC=1
      KLM=1
 38   LS=1
      LLL=2
      H=0.
      CALL ACCUM(2)
      GO TO(405,68,1003),LS
 405  NF=NF+1
      ND=ND+1
      F4=F3
      GO TO(43,499),NPH
 43   CALL BOUND(2,X)
 499  FMAX=F3
      F6=F7
      F5=F4
```

```
      GO TO(209,7),LCUT
209   WRITE(6,210)ND
210   FORMAT(10HOITERATIONI6)
      WRITE(6,207)F3,NF,(C1(I),I=1,L)
207   FORMAT(9HOFUNCTIONE17.7,13H    EVALUATIONI6/11H PARAMETERS/(7E17.7)
     2)
7     GO TO(742,736),KLM
742   GO TO(506,528,510),IND
528   DO 525 I=1,L
      DO 525 J=1,L
525   C(I,J)=G(I,J)
      CALL EIG(L,1)
      AAC=1.
      DO 526 I=1,L
      IF(C(I,I)-AL)530,530,526
526   AAC=AAC*C(I,I)
      IF(AAO-BL)530,530,510
530   GO TO(532,506),LCUT
532   WRITE(6,531)(C(I,I),I=1,L)
531   FORMAT(44HOMATRIX REINITIALIZED BECAUSE OF SINGULARITY/29H EIGENVA
     2LUES OF SCALED MATRIX/(7E16.6))
506   KLM=1
      DO 72 I=1,L
      DO 73 J=1,L
73    G(I,J)=0.
      IF(Y(I)*F(I))508,509,508
509   G(I,I)=1.
      GO TO 72
508   G(I,I)=ABS(Y(I)/F(I))
72    CONTINUE
510   IND=2
      DO 5 I=1,L
5     FF(I)=C1(I)
74    DF=0.
      DO 700 I=1,L
      Y(I)=0.
      DO 701 J=1,L
701   Y(I)=Y(I)+G(I,J)*F(J)
700   DF=DF+Y(I)*F(I)
      IF(DF)704,704,702
704   GO TO(901,19),KKK
901   KKK=2
      KLM=1
      GO TO 506
702   ALO=0.
      F8=F3
      AMX=0.
      KKK=1
      IF(NCON)600,600,601
600   KUT=1
      LLL=1
601   ALMAX=1.E30
24    FO=F3
      ALSM=1.E30
      DO 705 I=1,L
      IF(Y(I))707,705,707
707   ALSM=AMIN1(ALSM,EPS/ABS(Y(I))*(ABS(C1(I))+EPS1))
705   CONTINUE
401   AL1=AMIN1(2.*ABS(FO)/DF,.5*ALMAX)
      IJK=1
```

```
      ALMIN=AL1
      GO TO 800
803   FF1=F3
      AL1=ALMIN
709   ALMIN=-DF/2.*AL1/(FF1-F0-DF*AL1)*AL1
      IF(ALMIN)710,710,711
710   AL2=AMIN1(2.*AL1,AL1+.999*(ALMAX-AL1))
      IF(AL2-AL1-ALSM)716,716,712
712   IJK=2
      ALMIN=AL2
      GO TO 800
804   F2=F3
      AL2=ALMIN
      IF(F2-FF1)900,900,714
714   AL0=AL1
      F0=FF1
      AL1=AL2
      FF1=F2
715   IF(AL1-ALMAX)710,716,716
900   EPL=AMAX1(EPSIL*(AL2-AL0),ALSM)
      NME=1
713   AA0=F0-FF1
      AA2=F2-FF1
      GO TO(300,523),NME
300   IF(ABS(AA0/AA2)-10.)521,521,522
521   IF(ABS(AA2/AA0)-10.)523,523,524
522   ALMIN=AL0+.618*(AL1-AL0)
      GO TO 30
524   ALMIN=AL2-.618*(AL2-AL1)
      GO TO 30
523   AA0=(AL1-AL2)*AA0
      AA2=(AL0-AL1)*AA2
      NME=2
      ALMIN=(AA0*(AL1+AL2)+AA2*(AL0+AL1))/(AA0+AA2)*.5
30    IF(ABS(ALMIN-AL1)-EPL)716,716,717
717   IF(ALMIN-ALSM)808,808,809
809   IJK=3
      GO TO 800
805   IF(ALMIN-AL1)718,716,719
718   IF(F3-FF1)720,720,721
720   IF(F3-F0)744,744,745
744   IF(AL0)746,746,747
747   AL2=ALMIN
      F2=F3
      AL1=AL0
      FF1=F0
      AL0=0.
      F0=F8
      FMAX=FF1
      AMX=AL1
      GO TO 713
746   FMAX=F0
      AMX=0.
      GO TO 803
745   AL0=ALMIN
      F0=F3
      GO TO 713
721   AL2=AL1
      F2=FF1
722   AL1=ALMIN
```

```
      FF1=F3
      GO TO 713
719   IF(F3-FF1)723,723,724
723   AL2=ALMIN
      F2=F3
      GO TO 713
724   AL0=AL1
      F0=FF1
      GO TO 722
711   IF(ALMIN-AL1)725,710,726
726   AL2=AMIN1(ALMIN,.999*ALMAX)
      GO TO 712
725   IJK=4
      IF(ALMIN-ALSM)808,808,800
800   LS=1
      GO TO(602,603),LLL
603   IF(ALMIN-H)602,602,605
605   H=ALMIN
      CALL BOUND(4,H)
      IF(H-ALMIN)604,602,602
604   CALL BOUND(5,H)
      ALMAX=H
      LLL=1
      ALMIN=.9*ALMAX
602   DO 802 I=1,L
802   C1(I)=FF(I)+ALMIN*Y(I)
      CALL ACCUM(1)
      GO TO(807,402,1003),LS
402   ALMAX=ALMIN
      GO TO(401,500,716,716),IJK
500   ALMIN=.5*(AL1+ALMIN)
      IF(ALMIN-AL1-ALSM)716,716,800
807   NF=NF+1
      F4=F3
      GO TO(31,32),NPH
31    CALL BOUND(1,X)
32    GO TO(33,34),LOUT
33    WRITE(6,207)F3,NF,(C1(I),I=1,L)
34    IF(F3-FMAX)35,35,36
36    AMX=ALMIN
      FMAX=F3
      F6=F7
      F5=F4
35    GO TO(803,804,805,806),IJK
806   IF(F3-F0)727,727,728
727   AL1=ALMIN
      FF1=F3
      GO TO 709
728   IF(F3-FF1)729,730,730
729   AL0=ALMIN
      F0=F3
      GO TO 715
730   AL2=AL1
      F2=FF1
      AL1=ALMIN
      FF1=F3
      GO TO 900
716   F3=FMAX
      IF(AMX)808,808,42
42    DO 731 I=1,L
```

```
      Y(I)=Y(I)*AMX
      C1(I)=FF(I)+Y(I)
  731 FF(I)=F(I)
      KLM=2
      GO TC 38
  736 FF1=0.
      DC 738 I=1,L
      EGV(I)=F(I)-FF(I)
  738 FF1=FF1+Y(I)*EGV(I)
      F2=0.
      CO 739 I=1,L
      FF(I)=C.
      DO 74C J=1,L
  740 FF(I)=FF(I)+G(I,J)*EGV(J)
  739 F2=F2+EGV(I)*FF(I)
      DO 904 I=1,L
      DO 904 J=I,L
  904 G(I,J)=G(I,J)-FF(I)/F2*FF(J)-Y(I)/FF1*Y(J)
      CO 743 I=2,L
      CO 743 J=2,I
  743 G(I,J-1)=G(J-1,I)
   39 IF(AMX-ALSM)19,19,742
  8C8 F3=FMAX
      CO 810 I=1,L
  810 C1(I)=FF(I)
   19 GO TO(1002,1C03),NPH
 1002 KLM=1
      IND=3
      IF(IPH)1CC4,10C4,1CC5
 1004 NPH=2
      GC TO(121,38),LCUT
  121 WRITE(6,214)
  214 FORMAT(20HONC PENALTY FUNCTICN)
      GO TC 38
 1005 IF(ABS(F3-F5)-.1)1C04,10C6,1006
 1006 CALL BCUND(6,H)
      IPH=IPH-1
      GO TO(213,38),LCUT
  213 WRITE(6,211)
  211 FORMAT(41HOPENALTY FUNCTICN REDUCED BY FACTCR CF 1C)
      GC TC 38
 1003 WRITE(6,123)F3,(C1(I),I=1,L)
  123 FORMAT(30HOMAXIMUM OF CBJECTIVE FUNCTIONE17.7/11HOPARAMETERS/(7E17
     2.7))
      WRITE(6,9CC1)NF,ND
 9001 FCRMAT(21HOFUNCTICN EVALUATICNSI6,25H   DERIVATIVE EVALUATICNSI5)
      WRITE(6,920)(F(I),I=1,L)
  920 FORMAT(9HIGRADIENT/(7E17.6))
      WRITE(6,921)
  921 FORMAT(20HO-INVERSE CF HESSIAN)
      DC 922 I=1,L
  922 WRITE(6,923)(G(I,J),J=1,L)
  923 FCRMAT(7E17.6)
      DC 40 I=1,L
      DO 40 J=1,L
   40 C(I,J)=G(I,J)
      CALL EIG(L,2)
      WRITE(6,41)(C(I,I),I=1,L)
   41 FCRMAT(39HO-EIGENVALUES CF SCALED INVERSE HESSIAN/(7E17.6))
      WRITE(6,44)
```

```
44    FORMAT(21HOPRINCIPAL CCMPCNENTS)
      DO 45 I=1,L
45    WRITE(6,37)(GL(J,I),J=1,L)
   37 FORMAT (/7E17.6/(7E17.6))
      CALL BCUNC(7,H)
      F7=F6
      CALL CLT
907   RETURN
68    J=1
      DO 71 I=1,L
      Y(I)=.5*Y(I)
      IF(ABS(Y(I))/(EPS1+AES(C1(I)))-EPS)71,71,925
925   J=2
71    C1(I)=C1(I)-Y(I)
      GO TC(9C9,926),J
9C9   WRITE(6,910)
910   FORMAT(45HOFEASIBLE PARAMETER VALUES CCULD NCT BE FOUND/45H *****
     2****************************************)
      RETURN
926   CCNTINLE
      WRITE(6,924)
924   FCRMAT(8HORESTART)
      GO TO 38
      END
```

```
C DECK 13
C OUTPUT FOR NON-LINEAR PROGRAMMING PROBLEMS
      SUBROUTINE OUT
      RETURN
      END
```

```
C COMPUTE CCNSTRAINT FUNCTIONS AND THEIR FIRST DERIVATIVES
      SUBRCUTINE LIMIT (II,I,X)
      COMMON C(20,20),G1(20,20),PSCA,G(20,20),F(20),Y(20),EGV(20),FF(20
     1,TITLE(20),CUB(20),CLB(20),PNL(20),NCON,LOUT,F3,NTH,F6,F7,METH,NP
     2,MD,LS,C1(20)
      GO TO (1,1,2),II
    1 GO TO (3,4,5,6,7,8,9,10,11,12,13,14,20,21,22,23,24),I
C CCNSTRAINT--P1,P2,C1,C2,C3,C4 LARGE THAN ZERO AND SMALLER THAN OR
C    EQUIRE TO CNE
C ****************
C                  *
C   IN THIS PROGRAM *
C C1(1) = P1       *
C C1(2) = P2       *
C C1(3) = C1       *
C C1(4) = C2       *
C C1(5) = C3       *
C C1(6) = C4       *
C                  *
C ****************
    3 X=C1(1)-1.0
      GO TC(2,31),II
    4 X=-C1(1)
      GO TO (2,41),II
    5 X=C1(2)-1.0
      GO TO (2,51),II
    6 X=-C1(2)
      GO TO (2,61),II
    7 X=C1(3)-1.
      GO TO (2,71),II
    8 X=-C1(3)
      GO TC (2,81),II
    9 X=C1(4)-1.
      GO TO (2,91),II
   10 X=-C1(4)
      GO TO (2,101),II
   11 X=C1(5)-1.
      GO TC (2,111),II
   12 X=-C1(5)
      GO TO (2,121),II
   13 X=C1(6)-1.
      GO TO (2,131),II
   14 X=-C1(6)
      GO TC (2,141),II
C CCNSTRAINT FUNCTIONS   C1+C4=1   C2+C3=1  C1+C3=1  C1+C2=1  4P1+2P2=1
   20 X=C1(5)+C1(6)-1.
      GO TO(2,201),II
   21 X=C1(3)+C1(4)-1.
      GO TC(2,211),II
   22 X=C1(4)+C1(5)-1.
      GC TC(2,221),II
   23 X=C1(3)+C1(6)-1.
      GO TC(2,231),II
   24 X=4.*C1(1)+2.*C1(2)-1.
      GO TO(2,241),II
C THE FCLLCWING EQUATICNS ARE THE FIRST DERIVATIVE OF THE CCNSTRAINTS
C CUB(I) WHERE I=(1,6), IS THE FIRST DERIVATIVE CF    X=P1-1
C WITH RESPECT TO P1,P2,C1,C2,C3,C4 RESPECTIVELY
   31 CUB(1)=1.
      CUB(2)=0.
```

```
      CUB(3)=0.
      CUB(4)=C.
      CUB(5)=0.
      CUB(6)=0.
      GO TO 2
C CUB(I) WHERE I=(1,6), IS THE FIRST DERIVERTIVE OF     X=-P1
C WITH RESPECT TO P1,P2,C1,C2,C3,C4 RESPECTIVELY
   41 CUB(1)=-1.
      CUB(2)=0.
      CUB(3)=C.
      CUB(4)=0.
      CUB(5)=0.
      CUB(6)=0.
      GC TC 2
C CUB(I) WHERE I=(1,6), IS THE FIRST CERIVERTIVE OF     X=P2-1
C WITH RESPECT TO P1,P2,C1,C2,C3,C4 RESPECTIVELY
   51 CUB(1)=0.
      CUB(2)=1.
      CUB(3)=0.
      CUB(4)=0.
      CUB(5)=0.
      CUB(6)=0.
      GO TO 2
C CUB(I) WHERE I=(1,6), IS THE FIRST DERIVATIVE OF     X=-P2
C WITH RESPECT TO P1,P2,C1,C2,C3,C4 RESPECTIVELY
   61 CUB(1)=C.
      CUB(2)=-1.
      CUB(3)=0.
      CUB(4)=0.
      CUB(5)=0.
      CUB(6)=C.
      GO TO 2
C CUB(I) WHERE I=(1,6), IS THE FIRST CERIVATIVE CF     X=C1-1
C WITH RESPECT TO P1,P2,C1,C2,C3,C4 RESPECTIVELY
   71 CUB(1)=0.
      CUB(2)=0.
      CUB(3)=1.
      CUB(4)=0.
      CUB(5)=C.
      CUB(6)=0.
      GC TO 2
C CUB(I) WHERE I=(1,6), IS THE FIRST CERIVATIVE CF     X=-C1
C WITH RESPECT TO P1,P2,C1,C2,C3,C4 RESPECTIVELY
   81 CUB(1)=C.
      CUB(2)=C.
      CUB(3)=-1.
      CUB(4)=0.
      CUB(5)=0.
      CUB(6)=0.
      GO TC 2
C CUB(I) WHERE I=(1,6), IS THE FIRST DERIVATIVE CF     X=C2-1
C WITH RESPECT TO P1,P2,C1,C2,C3,C4 RESPECTIVELY
   91 CUB(1)=C.
      CUB(2)=0.
      CUB(3)=0.
      CUB(4)=1.
      CUB(5)=0.
      CUB(6)=C.
      GC TC 2
C CUB(I) WHERE I=(1,6), IS THE FIRST DERIVATIVE CF     X=-C2
```

```
C WITH RESPECT TO P1,P2,C1,C2,C3,C4 RESPECTIVELY
  101 CUB(1)=0.
      CUB(2)=0.
      CUB(3)=0.
      CUB(4)=-1.
      CUB(5)=0.
      CUB(6)=0.
      GO TO 2
C CUB(I) WHERE I=(1,6), IS THE FIRST DERIVATIVE OF    X=C3-1
C WITH RESPECT TO P1,P2,C1,C2,C3,C4 RESPECTIVELY
  111 CUB(1)=0.
      CUB(2)=0.
      CUB(3)=0.
      CUB(4)=0.
      CUB(5)=1.
      CUB(6)=0.
      GO TO 2
C CUB(I) WHERE I=(1,6), IS THE FIRST DERIVATIVE OF    X=-C3
C WITH RESPECT TO P1,P2,C1,C2,C3,C4 RESPECTIVELY
  121 CUB(1)=0.
      CUB(2)=0.
      CUB(3)=0.
      CUB(4)=0.
      CUB(5)=-1.
      CUB(6)=0.
      GO TO 2
C CUB(I) WHERE I=(1,6), IS THE FIRST DERIVATIVE OF    X=C4-1
C WITH RESPECT TO P1,P2,C1,C2,C3,C4 RESPECTIVELY
  131 CUB(1)=0.
      CUB(2)=0.
      CUB(3)=0.
      CUB(4)=0.
      CUB(5)=0.
      CUB(6)=1.
      GO TO 2
C CUB(I) WHERE I=(1,6), IS THE FIRST DERIVATIVE OF    X=-C4
C WITH RESPECT TO P1,P2,C1,C2,C3,C4 RESPECTIVELY
  141 CUB(1)=0.
      CUB(2)=0.
      CUB(3)=0.
      CUB(4)=0.
      CUB(5)=0.
      CUB(6)=-1.
      GO TO 2
C CUB(I) WHERE I=(1,6), IS THE FIRST DERIVATIVE OF  X=C3+C4-1
C WITH RESPECT TO P1,P2,C1,C2,C3,C4 RESPECTIVELY
  201 CUB(1)=0.
      CUB(2)=0.
      CUB(3)=0.
      CUB(4)=0.
      CUB(5)=1.
      CUB(6)=1.
      GO TO 2
C CUB(I) WHERE I=(1,6), IS THE FIRST DERIVATIVE OF  X=C1+C2-1
C WITH RESPECT TO P1,P2,C1,C2,C3,C4 RESPECTIVELY
  211 CUB(1)=0.
      CUB(2)=0.
      CUB(3)=1.
      CUB(4)=1.
      CUB(5)=0.
```

```
      CUB(6)=C.
      GO TC 2
C CUB(I) WHERE I=(1,6), IS THE FIRST CERIVERTIVE OF X=C2+C3-1
C WITH RESPECT TO P1,P2,C1,C2,C3,C4 RESPECTIVELY
  221 CUB(1)=0.
      CUB(2)=0.
      CUB(3)=0.
      CUB(4)=1.
      CUB(5)=1.
      CUB(6)=0.
      GO TC 2
C CUB(I) WHERE I=(1,6), IS THE FIRST DERIVATIVE CF  X=C1+C4-1
C WITH RESPECT TC P1,P2,C1,C2,C3,C4 RESPECTIVELY
  231 CUB(1)=0.
      CUB(2)=0.
      CUB(3)=1.
      CUB(4)=0.
      CUB(5)=0.
      CUB(6)=1.
      GO TC 2
C CUB(I) WHERE I=(1,6), IS THE FIRST DERIVATIVE CF  X=4P1+2P2-1
C WITH RESPECT TC P1,P2,C1,C2,C3,C4 RESPECTIVELY
  241 CUB(1)=4.
      CUB(2)=2.
      CUB(3)=0.
      CUB(4)=C.
      CUB(5)=0.
      CUB(6)=C.
    2 RETURN
      END
```

```
C COMPUTES VALUE OF OBJECTIVE FUNCTION AND ITS DERIVATIVE
C THE FIRST METHOD---- WITH REPETITIVE TRANSITION
      SUBROUTINE ACCUM(II)
      COMMON C(20,20),C1(20,20),PSCA,G(20,20),F(20),Y(20),EGV(20),FF(20
     1,TITLE(20),CUB(20),CLB(20),PNL(20),NCON,LOUT,F3,NTH,F6,F7,METH,NP
     2,MC,LS,C1(20)
      DIMENSION YTH(8,8)
      GO TO (1,1,2),II
C ******************
C                  *
C  IN THIS PROGRAM *
C C1(1) = P1       *
C C1(2) = P2       *
C C1(3) = C1       *
C C1(4) = C2       *
C C1(5) = C3       *
C C1(6) = C4       *
C                  *
C ******************
C  T REPRESENT THE PROBABILITY OF START AT STATE I AND RETURNING TO I
C AFTER HAVING TRAVERSED ALL POSSIBLE LOOPS FROM ZERO TO AN INFINITE
C NUMBER OF TIMES.
    1 T=1./((1.-C1(2)*C1(4)*C1(5))**2*(1.-C1(2)*C1(3)*C1(6))**2
     1*(1.-C1(1)*C1(3))*(1.-C1(1)*C1(4))*(1.-C1(1)*C1(5))*(1.-C1(1)*
     2C1(6)))
      Z1=1.0-4.*C1(1)-2.*C1(2)
      Z2=1.0-C1(5)-C1(6)
      Z3=1.0-C1(3)-C1(4)
      Z4=1.0-C1(3)-C1(4)
      Z5=1.0-C1(5)-C1(6)
      Z6=1.0-C1(4)-C1(5)
      Z7=1.0-C1(3)-C1(6)
      E2=C1(1)+C1(2)*C1(4)
      E3=C1(1)+C1(2)*C1(5)
      E4=C1(1)+C1(2)*C1(6)
      E5=C1(1)+C1(2)*C1(3)
      E6=C1(2)
      E7=C1(2)
      E8=C1(1)*C1(4)+C1(1)*C1(6)+2.0*C1(2)*C1(4)*C1(6)
      E9=C1(1)*C1(3)+C1(1)*C1(5)+2.0*C1(2)*C1(3)*C1(5)
C THE OBJECTIVE FUNCTIONS AND THE GIVEN DATA
      Y1=Z1*T-0.9466
      Y2=T*(Z2*E2+Z3*E3+Z4*E4+Z5*E5)-0.0079
      Y3=T*(Z6*E6+Z7*E7)-0.0001
      Y4=T*(E8+E9)-0.0454
      A=-1.0
C THE OBJECTIVE -- SUMMING THE SQUARE  OF THE ABOVE FOUR EQUATIONS
C AND MINIMIZING THEM
      F3=(Y1**2+Y2**2+Y3**2+Y4**2)*A
      GO TO(2,3),II
    3 DTP1=T*(C1(3)/(1.-C1(1)*C1(3))+C1(4)/(1.-C1(1)*C1(4))+
     1C1(5)/(1.-C1(1)*C1(5))+C1(6)/(1.-C1(1)*C1(6)))
      DTP2=T*(2.*C1(4)*C1(5)/(1.-C1(2)*C1(4)*C1(5))+2.*C1(3)*C1(6)/
     1(1.-C1(2)*C1(3)*C1(6)))
      DTC1=T*(2.*C1(2)*C1(6)/(1.-C1(2)*C1(3)*C1(6))+C1(1)/
     1(1.-C1(1)*C1(3)))
      DTC2=T*(2.*C1(2)*C1(5)/(1.-C1(2)*C1(4)*C1(5))+C1(1)/
     1(1.-C1(1)*C1(4)))
      DTC3=T*(2.*C1(2)*C1(4)/(1.-C1(2)*C1(4)*C1(5))+C1(1)/
     1(1.-C1(1)*C1(5)))
```

```
      DTC4=T*(2.*C1(2)*C1(3)/(1.-C1(2)*C1(3)*C1(6))+C1(1)/
     1(1.-C1(1)*C1(6)))
      YTH(1,1)=DTP1*Z1-T*4.
      YTH(1,2)=DTP2*Z1-T*2.
      YTH(1,3)=DTC1*Z1
      YTH(1,4)=DTC2*Z1
      YTH(1,5)=DTC3*Z1
      YTH(1,6)=DTC4*Z1
      YTH(2,1)=DTP1*(Z2*E2+Z3*E3+Z4*E4+Z5*E5)+T*(Z2+Z3+Z4+Z5)
      YTH(2,2)=DTP2*(Z2*E2+Z3*E3+Z4*E4+Z5*E5)+T*(Z2*C1(3)+Z3*C1(5)+Z4*
     1C1(6)+Z5*C1(3))
      YTH(2,3)=DTC1*(Z2*E2+Z3*E3+Z4*E4+Z5*E5)+T*(Z5*C1(2)-E3-E4)
      YTH(2,4)=DTC2*(Z2*E2+Z3*E3+Z4*E4+Z5*E5)+T*(Z2*C1(2)-E3-E4)
      YTH(2,5)=DTC3*(Z2*E2+Z3*E3+Z4*E4+Z5*E5)+T*(Z3*C1(2)-E2-E5)
      YTH(2,6)=DTC4*(Z2*E2+Z3*E3+Z4*E4+Z5*E5)+T*(Z4*C1(2)-E5-E2)
      YTH(3,1)=DTP1*(Z6*E6+Z7*E7)
      YTH(3,2)=DTP2*(Z6*E6+Z7*E7)+T*(Z6+Z7)
      YTH(3,3)=DTC1*(Z6*E6+Z7*E7)+T*(-E7)
      YTH(3,4)=DTC2*(Z6*E6+Z7*E7)+T*(-E6)
      YTH(3,5)=DTC3*(Z6*E6+Z7*E7)+T*(-E6)
      YTH(3,6)=DTC4*(Z6*E6+Z7*E7)+T*(-E7)
      YTH(4,1)=DTP1*(E8+E9)+T*(C1(4)+C1(6)+C1(3)+C1(5))
      YTH(4,2)=DTP2*(E8+E9)+T*(2.0*C1(4)*C1(6)+2.0*C1(3)*C1(5))
      YTH(4,3)=DTC1*(E8+E9)+T*(C1(1)+2.0*C1(2)*C1(5))
      YTH(4,4)=DTC2*(E8+E9)+T*(C1(1)+2.0*C1(2)*C1(6))
      YTH(4,5)=DTC3*(E8+E9)+T*(C1(1)+2.0*C1(2)*C1(3))
      YTH(4,6)=DTC4*(E8+E9)+T*(C1(1)+2.0*C1(2)*C1(4))
C THE FIRST DERIVATIVE OF T WITH RESPECT TO P1,P2,C1,C2,C3,C4
C RESPECTIVELY
      F(1)=(2.*Y1*YTH(1,1)+2.*Y2*YTH(2,1)+2.*Y3*YTH(3,1)+2.*Y4*YTH(4,1))
     1*A
      F(2)=(2.*Y1*YTH(1,2)+2.*Y2*YTH(2,2)+2.*Y3*YTH(3,2)+2.*Y4*YTH(4,2))
     1*A
      F(3)=(2.*Y1*YTH(1,3)+2.*Y2*YTH(2,3)+2.*Y3*YTH(3,3)+2.*Y4*YTH(4,3))
     1*A
      F(4)=(2.*Y1*YTH(1,4)+2.*Y2*YTH(2,4)+2.*Y3*YTH(3,4)+2.*Y4*YTH(4,4))
     1*A
      F(5)=(2.*Y1*YTH(1,5)+2.*Y2*YTH(2,5)+2.*Y3*YTH(3,5)+2.*Y4*YTH(4,5))
     1*A
      F(6)=(2.*Y1*YTH(1,6)+2.*Y2*YTH(2,6)+2.*Y3*YTH(3,6)+2.*Y4*YTH(4,6))
     1*A
    2 RETURN
      END
```

```
C CCMPUTES VALUE CF CBJECTIVE FUNCTICN ANC ITS DERIVATIVE
C THE SECOND METHCD---- NCRMALIZATICN CF THE FIRST METHOC.
      SUBRCUTINE ACCUM(II)
      COMMCN C(20,20),G1(2C,20),PSCA,G(20,20),F(20),Y(20),EGV(20),FF(20)
     1,TITLE(20),CUB(2C),CLB(2C),PNL(2C),NCON,LCUT,F3,NTH,F6,F7,METH,NPH
     2,MD,LS,C1(20)
      DIMENSICN YTH(8,8)
      GO TO (1,1,2),II
C *****************
C                  *
C  IN THIS PROGRAM *
C C1(1) = P1       *
C C1(2) = P2       *
C C1(3) = C1       *
C C1(4) = C2       *
C C1(5) = C3       *
C C1(6) = C4       *
C                  *
C *****************
C  T REPRESENT THE PROBABILITY OF START AT STATE I AND RETURNING TC I
C AFTER HAVING TRAVERSED ALL PCSSIBLE LOCPS FROM ZERO TO AN INFINITE
C NUMBER OF TIMES.
    1 T=1./((1.-C1(2)*C1(4)*C1(5))**2*(1.-C1(2)*C1(3)*C1(6))**2
     1*(1.-C1(1)*C1(3))*(1.-C1(1)*C1(4))*(1.-C1(1)*C1(5))*(1.-C1(1)*
     2C1(6)))
      Z1=1.0-4.*C1(1)-2.*C1(2)
      Z2=1.0-C1(5)-C1(6)
      Z3=1.C-C1(3)-C1(4)
      Z4=1.0-C1(3)-C1(4)
      Z5=1.C-C1(5)-C1(6)
      Z6=1.0-C1(4)-C1(5)
      Z7=1.0-C1(3)-C1(6)
      E2=C1(1)+C1(2)*C1(4)
      E3=C1(1)+C1(2)*C1(5)
      E4=C1(1)+C1(2)*C1(6)
      E5=C1(1)+C1(2)*C1(3)
      E6=C1(2)
      E7=C1(2)
      E8=C1(1)*C1(4)+C1(1)*C1(6)+2.0*C1(2)*C1(4)*C1(6)
      E9=C1(1)*C1(3)+C1(1)*C1(5)+2.0*C1(2)*C1(3)*C1(5)
C THE CBJECTIVE FUNCTICNS ANC THE GIVEN DATA
      Y1=Z1*T/C.9466-1.
      Y2=T*(Z2*E2+Z3*E3+Z4*E4+Z5*E5)/0.0079-1.
      Y3=T*(Z6*E6+Z7*E7)/0.0001-1.
      Y4=T*(E8+E9)/0.0454-1.
      A=-1.C
C THE CBJECTIVE -- SUMMING THE SQUARE  OF THE ABCVE FOUR EQUATIONS
C AND MINIMIZING THEM
      F3=(Y1**2+Y2**2+Y3**2+Y4**2)*A
      GO TC(2,3),II
    3 S1=1./C.9466
      S2=1./C.CC79
      S3=1./C.CC01
      S4=1./0.C454
      DTP1=T*(C1(3)/(1.-C1(1)*C1(3))+C1(4)/(1.-C1(1)*C1(4))+
     1C1(5)/(1.-C1(1)*C1(5))+C1(6)/(1.-C1(1)*C1(6)))
      DTP2=T*(2.*C1(4)*C1(5)/(1.-C1(2)*C1(4)*C1(5))+2.*C1(3)*C1(6)/
     1(1.-C1(2)*C1(3)*C1(6)))
      DTC1=T*(2.*C1(2)*C1(6)/(1.-C1(2)*C1(3)*C1(6))+C1(1)/
     1(1.-C1(1)*C1(3)))
```

```
      DTC2=T*(2.*C1(2)*C1(5)/(1.-C1(2)*C1(4)*C1(5))+C1(1)/
     1(1.-C1(1)*C1(4)))
      DTC3=T*(2.*C1(2)*C1(4)/(1.-C1(2)*C1(4)*C1(5))+C1(1)/
     1(1.-C1(1)*C1(5)))
      DTC4=T*(2.*C1(2)*C1(3)/(1.-C1(2)*C1(3)*C1(6))+C1(1)/
     1(1.-C1(1)*C1(6)))
      YTH(1,1)=(DTP1*Z1-T*4.)*S1
      YTH(1,2)=(DTP2*Z1-T*2.)*S1
      YTH(1,3)=DTC1*Z1*S1
      YTH(1,4)=DTC2*Z1*S1
      YTH(1,5)=DTC3*Z1*S1
      YTH(1,6)=DTC4*Z1*S1
      YTH(2,1)=DTP1*(Z2*E2+Z3*E3+Z4*E4+Z5*E5)+T*(Z2+Z3+Z4+Z5)
      YTH(2,1)=YTH(2,1)*S2
      YTH(2,2)=DTP2*(Z2*E2+Z3*E3+Z4*E4+Z5*E5)+T*(Z2*C1(3)+Z3*C1(5)+Z4*
     1C1(6)+Z5*C1(3))
      YTH(2,2)=YTH(2,2)*S2
      YTH(2,3)=DTC1*(Z2*E2+Z3*E3+Z4*E4+Z5*E5)+T*(Z5*C1(2)-E3-E4)
      YTH(2,3)=YTH(2,3)*S2
      YTH(2,4)=DTC2*(Z2*E2+Z3*E3+Z4*E4+Z5*E5)+T*(Z2*C1(2)-E3-E4)
      YTH(2,4)=YTH(2,4)*S2
      YTH(2,5)=DTC3*(Z2*E2+Z3*E3+Z4*E4+Z5*E5)+T*(Z3*C1(2)-E2-E5)
      YTH(2,5)=YTH(2,5)*S2
      YTH(2,6)=DTC4*(Z2*E2+Z3*E3+Z4*E4+Z5*E5)+T*(Z4*C1(2)-E5-E2)
      YTH(2,6)=YTH(2,6)*S2
      YTH(3,1)=DTP1*(Z6*E6+Z7*E7)*S3
      YTH(3,2)=DTP2*(Z6*E6+Z7*E7)+T*(Z6+Z7)
      YTH(3,2)=YTH(3,2)*S3
      YTH(3,3)=DTC1*(Z6*E6+Z7*E7)+T*(-E7)
      YTH(3,3)=YTH(3,3)*S3
      YTH(3,4)=DTC2*(Z6*E6+Z7*E7)+T*(-E6)
      YTH(3,4)=YTH(3,4)*S3
      YTH(3,5)=DTC3*(Z6*E6+Z7*E7)+T*(-E6)
      YTH(3,5)=YTH(3,5)*S3
      YTH(3,6)=DTC4*(Z6*E6+Z7*E7)+T*(-E7)
      YTH(3,6)=YTH(3,6)*S3
      YTH(4,1)=DTP1*(E8+E9)+T*(C1(4)+C1(6)+C1(3)+C1(5))
      YTH(4,1)=YTH(4,1)*S4
      YTH(4,2)=DTP2*(E8+E9)+T*(2.0*C1(4)*C1(6)+2.0*C1(3)*C1(5))
      YTH(4,2)=YTH(4,2)*S4
      YTH(4,3)=DTC1*(E8+E9)+T*(C1(1)+2.0*C1(2)*C1(5))
      YTH(4,3)=YTH(4,3)*S4
      YTH(4,4)=DTC2*(E8+E9)+T*(C1(1)+2.0*C1(2)*C1(6))
      YTH(4,4)=YTH(4,4)*S4
      YTH(4,4)=(C1(1)+2.*C1(2)*C1(6))*S4
      YTH(4,5)=DTC3*(E8+E9)+T*(C1(1)+2.0*C1(2)*C1(3))
      YTH(4,5)=YTH(4,5)*S4
      YTH(4,6)=DTC4*(E8+E9)+T*(C1(1)+2.0*C1(2)*C1(4))
      YTH(4,6)=YTH(4,6)*S4
C THE FIRST DERIVATIVE OF T WITH RESPECT TO P1,P2,C1,C2,C3,C4
C RESPECTIVELY
      F(1)=(2.*Y1*YTH(1,1)+2.*Y2*YTH(2,1)+2.*Y3*YTH(3,1)+2.*Y4*YTH(4,1))
     1*A
      F(2)=(2.*Y1*YTH(1,2)+2.*Y2*YTH(2,2)+2.*Y3*YTH(3,2)+2.*Y4*YTH(4,2))
     1*A
      F(3)=(2.*Y1*YTH(1,3)+2.*Y2*YTH(2,3)+2.*Y3*YTH(3,3)+2.*Y4*YTH(4,3))
     1*A
      F(4)=(2.*Y1*YTH(1,4)+2.*Y2*YTH(2,4)+2.*Y3*YTH(3,4)+2.*Y4*YTH(4,4))
     1*A
      F(5)=(2.*Y1*YTH(1,5)+2.*Y2*YTH(2,5)+2.*Y3*YTH(3,5)+2.*Y4*YTH(4,5))
```

```
 1*A
  F(6)=(2.*Y1*YTH(1,6)+2.*Y2*YTH(2,6 )+2.*Y3*YTH( 3,6)+2.*Y4*YTH(4,6))
 1*A
2 RETURN
  END
```

```
C CCMPUTES VALUE OF OBJECTIVE FUNCTICN AND ITS DERIVATIVE
C THE THIRD METHOD SIMPLIFICATION WITHOUT REPETITIVE TRANSITICN.
      SUBROUTINE ACCUM(II)
      COMMON C(20,20),G1(20,20),PSCA,G(20,20),F(20),Y(20),EGV(20),FF(20)
     1,TITLE(20),CUB(20),CLB(20),PNL(20),NCON,LOUT,F3,NTH,F6,F7,METH,NPH
     2,MD,LS,C1(20)
      DIMENSICN YTH(8,8)
C *****************
C                   *
C  IN THIS PROGRAM *
C C1(1) = P1        *
C C1(2) = P2        *
C C1(3) = C1        *
C C1(4) = C2        *
C C1(5) = C3        *
C C1(6) = C4        *
C                   *
C *****************
      GO TC (1,1,2),II
C FUNCTION T IS THE PRCBABILITY CF START STATE I TC ITSELF
    1 T=(2.*C1(2)*(C1(4)*C1(5)+C1(3)*C1(6))+C1(1)*
     1(C1(3)+C1(4)+C1(5)+C1(6))+1.)
      Z1=1.C-4.*C1(1)-2.*C1(2)
      Z2=1.0-C1(5)-C1(6)
      Z3=1.C-C1(3)-C1(4)
      Z4=1.0-C1(3)-C1(4)
      Z5=1.0-C1(5)-C1(6)
      Z6=1.0-C1(4)-C1(5)
      Z7=1.0-C1(3)-C1(6)
      E2=C1(1)+C1(2)*C1(4)
      E3=C1(1)+C1(2)*C1(5)
      E4=C1(1)+C1(2)*C1(6)
      E5=C1(1)+C1(2)*C1(3)
      E6=C1(2)
      E7=C1(2)
      E8=C1(1)*C1(4)+C1(1)*C1(6)+2.0*C1(2)*C1(4)*C1(6)
      E9=C1(1)*C1(3)+C1(1)*C1(5)+2.0*C1(2)*C1(3)*C1(5)
C THE CBJECTIVE FUNCTICNS ANC THE GIVEN CATA
      Y1=Z1*T/C.9466-1.
      Y2 = (Z2*E2+Z3*E3+Z4*E4+Z5*E5)/0.0079-1.
      Y3 = (Z6*E6+Z7*E7)/0.0001-1.
      Y4 = (E8+E9)/0.0454-1.
      A=-1.0
C THE CBJECTIVE -- SUMMING THE SQUARE  OF THE ABCVE FOUR EQUATICNS
C AND MINIMIZING THEM
      F3=(Y1**2+Y2**2+Y3**2+Y4**2)*A
      GO TO(2,3),II
    3 S1=1./C.9466
      S2=1./0.0079
      S3=1./C.0001
      S4=1./0.0454
      DTP1=C1(3)+C1(4)+C1(5)+C1(6)
      DTP2=2.*(C1(4)*C1(5)+C1(3)*C1(6))
      DTC1 =(2.*C1(2)*C1(6)+C1(1))
      DTC2= (2.*C1(2)*C1(5)+C1(1))
      DTC3= (2.*C1(2)*C1(4)+C1(1))
      DTC4= (2.*C1(2)*C1(3)+C1(1))
      YTH(1,1)=(DTP1*Z1-T*4.)*S1
      YTH(1,2)=(DTP2*Z1-T*2.)*S1
      YTH(1,3)=DTC1*Z1*S1
```

```
      YTH(1,4)=DTC2*Z1*S1
      YTH(1,5)=DTC3*Z1*S1
      YTH(1,6)=DTC4*Z1*S1
      YTH(2,1)=(Z2+Z3+Z4+Z5)*S2
      YTH(2,2)=(Z2*C1(3)+Z3*C1(5)+Z4*C1(6)+Z5*C1(3))*S2
      YTH(2,3)=(Z5*C1(2)-E3-E4)*S2
      YTH(2,4)=(Z2*C1(2)-E3-E4)*S2
      YTH(2,5)=(Z3*C1(2)-E2-E5)*S2
      YTH(2,6)=(Z4*C1(2)-E2-E5)*S2
      YTH(3,1)=0.
      YTH(3,2)=(Z6+Z7)*S3
      YTH(3,3)=(-E7)*S3
      YTH(3,4)=(-E6)*S3
      YTH(3,5)=(-E6)*S3
      YTH(3,6)=YTH(3,3)
      YTH(4,1)=(C1(4)+C1(6)+C1(3)+C1(5))*S4
      YTH(4,2)=(2.*C1(4)*C1(6)+2.*C1(3)*C1(5))*S4
      YTH(4,3)=(C1(1)+2.*C1(2)*C1(5))*S4
      YTH(4,5)=(C1(1)+2.*C1(2)*C1(3))*S4
      YTH(4,6)=(C1(1)+2.*C1(2)*C1(4))*S4
C THE FIRST DERIVATIVE OF T WITH RESPECT TO P1,P2,C1,C2,C3,C4
C RESPECTIVELY
      F(1)=(2.*Y1*YTH(1,1)+2.*Y2*YTH(2,1)+2.*Y3*YTH(3,1)+2.*Y4*YTH(4,1))
     1*A
      F(2)=(2.*Y1*YTH(1,2)+2.*Y2*YTH(2,2)+2.*Y3*YTH(3,2)+2.*Y4*YTH(4,2))
     1*A
      F(3)=(2.*Y1*YTH(1,3)+2.*Y2*YTH(2,3)+2.*Y3*YTH(3,3)+2.*Y4*YTH(4,3))
     1*A
      F(4)=(2.*Y1*YTH(1,4)+2.*Y2*YTH(2,4)+2.*Y3*YTH(3,4)+2.*Y4*YTH(4,4))
     1*A
      F(5)=(2.*Y1*YTH(1,5)+2.*Y2*YTH(2,5)+2.*Y3*YTH(3,5)+2.*Y4*YTH(4,5))
     1*A
      F(6)=(2.*Y1*YTH(1,6)+2.*Y2*YTH(2,6)+2.*Y3*YTH(3,6)+2.*Y4*YTH(4,6))
     1*A
    2 RETURN
      END
```

References

(1) Booth, Taylor L. (1968) Sequential Machines and Automata Theory, John Wiley and Sons Inc., N.Y.

(2) Paz, Azaria (1971) Introduction to Probabilistic Automata Academic Press N.Y.

(3) Mortimer, R. (1975) Private Ammunication.

(4) Hopcroft, John E. and Ullman, Jeffrey D. (1969) Formal Language and Their Relation to Automata, Addison-Wesely Publishing Company, Menlo Park, California.

(5) Bard, Yonathan (1967) Nonlinear Parameter Estimation and Programming IBM N.Y. Scientific Center, N.Y.

(6) Maryanski, Fred J. (1974) Inference of Probabilistic Grammars, Ph.D. Thesis, Department of Electrical Engineering (Computer Science), The University of Connecticut.

PARAMETER ESTIMATION OF A PROBABILISTIC AUTOMATA MODEL OF DNA MEIOSIS

BY

KUANG CHAN WU

B.S., Chung Cheng Institute of Technology, Taiwan, 1963.

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements of the degree

MASTER OF SCIENCE

Department of Computer Science

Kansas State University
Manhattan, Kansas

1976.

Abstract

The purpose of this report is to estimate parameters of a model to fit sample data as closely as possible. The model is of DNA (Deoxynucleic Acid) Meiosis. The sample data are taken from the work of Reference (3).

The models of DNA can be represented as a probabilistic Finite-state automata. In this report, properties and rules of both probabilistic finite state automata and probabilistic regular grammars were employed to developed the equations of the given model. A nonlinear programming technique was used to estimate parameters of these equations from experimental data.