

01170

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO



DIVISIÓN DE ESTUDIOS DE POSTGRADO,
FACULTAD DE INGENIERÍA

**MODELACIÓN DE UNA POBLACIÓN DE VIH
MEDIANTE ALGORITMOS GENÉTICOS**

TRABAJO DE TESIS QUE PARA OBTENER EL GRADO DE
MAESTRÍA EN INGENIERÍA INFORMÁTICA
(ELECTRICA)

PRESENTA:

SARA VERA NOGUEZ

DIRECTOR DE TESIS:
DR. HENRI WELBROECK G.

CIUDAD UNIVERSITARIA, D.F., MEXICO, OCTUBRE DE 1997.

TESIS CON
FALLA DE ORIGEN

1997



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Los miembros del jurado:

Dr. Christopher Stephens S.

Dr. Henri Waelbroeck G.

Dr. Felipe Lara Rosano.

Dr. Juan Carlos Almagro D.

A: Una gran mujer muestra de
sabiduría, entrega y amor.

Sara Noguez Saldaña

! -
-
-
Mi agradecimiento a todas aquellas personas e instituciones que colaboraron en la realización del presente trabajo, muy especialmente al Ing. Sergio Beltran López, fundador e impulsor de la Maestría en Ingeniería Informática, al Dr. Henri Waelbroeck por todo el trabajo y tiempo dedicado a la formación de quienes tuvimos la oportunidad de enriquecernos con sus conocimientos, al Dr. Christopher Stephens y al M en I. Ismael Arcos Q. por su paciente y minuciosa revisión así como por sus valiosas sugerencias; al M. en I Odón de Jesús Angeles Palacios, al Dr. Juan Carlos Almagro D., a Juanita Cruz, Beatriz Vera, Alvaro Vera, al Instituto de Ciencias Nucleares de la UNAM, y a Bisíntesis Laboratorios. Y muy especialmente a Francisco Chavez Castañeda.

-
-
-
Sara Vera Noguez.

Contenido

Introducción

1. Antecedentes biológicos

Introducción

Los virus y su reproducción

La respuesta inmune

El virus de la inmunodeficiencia humana (VIH)

2. Modelo matemático para la población de VIH

3. El modelo genético

Introducción

Codificación (genotipos y fenotipos)

Intérprete

Función de aptitud NK y su topología (fitness NK)

Algoritmo de Kauffman

El modelo para el VIH (algoritmo genético [AG] para el VIH *in vivo*)

Ecuaciones iterativas

Comparación del AG con el modelo de Nowak

Análisis de coeficientes de Liapunov

4. Resultados

Casos sin células T, sin V, sin células infectadas

Regímenes dinámicos

Recuperación

Latencia prolongada

Colapso del sistema inmune

Comentarios

Apéndice A. Código del algoritmo genético

Apéndice B. Reglas booleanas

Referencias

Introducción

En este trabajo se presenta un modelo de población del virus de la inmunodeficiencia humana (VIH) en un paciente *in vivo*. Para el desarrollo de este trabajo, se analizó el problema, se estudió un modelo matemático, se propuso un algoritmo genético que refleja su comportamiento, y finalmente se analizaron los resultados obtenidos. De esta forma se cubren los pasos de todo proceso de modelación basados en el conocimiento: valoración, aplicación y avance.

El trabajo está organizado en cinco capítulos: el primero es una breve introducción de Biología e Inmunología, en el que se describe cómo se lleva a cabo la respuesta inmune en un organismo, así como la definición de un virus, su estructura y reproducción; en el segundo, se expone el modelo matemático que representa el crecimiento de la población de VIH *in vivo*, explicado fisiológicamente; en el tercero, se da una breve introducción a algoritmos genéticos (AG), para finalmente plantear y explicar este modelo. Se incluye también una sección de pruebas, resultados y conclusiones en el capítulo cuarto. Los detalles de la programación del algoritmo son presentados en el apéndice A.

La modelación ha facilitado la solución de diversos problemas (v.g. problemas de estrategias de organización, social, de toma de decisiones, de finanzas; así como fenómenos mecánicos, termodinámicos, biológicos, etc.), ya que mediante el uso de modelos se pueden prever situaciones inconvenientes, optimar recursos y tener una mejor idea del comportamiento de los sistemas que se modelan.

Un modelo, nos dice C.M. Newton [C. Nicolini, 1984], es una representación formal de una hipótesis que marca un dominio de observación. Según este autor, los modelos son empíricos cuando se basan en la identificación de funciones observadas en datos; si el modelo refleja los factores físicos o biológicos más importantes en la observación de un fenómeno, será un modelo físico o biológico.

A partir de los modelos, es posible obtener fórmulas del comportamiento del sistema, mayor información, y más precisa, además de que facilitan el diseño de experimentos.

En general, los modelos ayudan a guiar, interpretar y aplicar pruebas en un sistema; su aplicación es tal que normalmente construimos modelos

mentales de todos los procesos o interacciones que nos interesan, a fin de comprenderlos.

Existen varias herramientas, o métodos de modelación —como las ecuaciones diferenciales, algoritmos genéticos, programación lineal y no lineal, métodos estocásticos y deterministas, etc.—, cuya selección depende del tipo de problema a resolver. Por ejemplo, si clasificamos los sistemas en continuos y discretos, las ecuaciones diferenciales son aplicables en los primeros; mientras que los AG en los segundos; por otra parte, si la clasificación distinguiera entre determinísticos y estocásticos, entonces, mediante una función, se puede encontrar el valor exacto que determine el estado del sistema, o utilizar un método que lo aproxime; etc.

Como se mencionó anteriormente, este trabajo se basa en un modelo de población, por lo que nos enfocaremos a este tipo de problemas. Con los modelos de cambio en poblaciones biológicas y en ecosistemas se pretende representar las características más importantes de un individuo en un proceso complejo [S.A. Levin, A. S. Perlson *et al.*, 1997]. Para ello se necesita entender las interacciones entre individuos, discernir entre datos y ruido, y comprender a cada individuo como una biosfera en miniatura con una escala de tiempo pequeña. En el modelo se debe reflejar la relación entre codificación y función de cada individuo, evolución, cooperatividad, altruismo, competencia, etc.

Un modelo puede reflejar el comportamiento de varios sistemas, por ejemplo,

$$\frac{dx(t)}{dt} = rx(t) \quad \text{representa el crecimiento de la población } x \text{ de una}$$

cierta especie, siendo el parámetro r la tasa de crecimiento. Frecuentemente, los parámetros del modelo se ajustan de forma empírica, basándose en la optimización de una función sobre el espacio de configuraciones del sistema en particular, conocido como topografía o *landscape*. Esta técnica empírica es fácilmente aplicable, pero sólo para problemas en los que se cuente con una función de optimización fija.

En los llamados “*sistemas adaptativos*” no se cuenta con una función de optimización fija. La evaluación de una configuración depende del entorno, definido por una “población” de otras configuraciones coexistentes. Por ejemplo, en los *sistemas genéticos* —cuyas configuraciones se llaman individuos— el desempeño de un individuo depende de la presencia de otros individuos en el momento. Otro ejemplo es el de los *mercados financieros*, en

donde un agente usa ciertas reglas para decidir sus posturas de compra y venta, pero su desempeño depende de las acciones de todos los demás participantes en el mercado.

En este trabajo se modela un sistema constituido por dos "tipos" de individuos que compiten y coexisten. En estas condiciones, no se puede definir una función de optimización, como en la mayoría de los casos de la naturaleza, sino que el mismo sistema determina el comportamiento mediante la competencia entre especies.

Específicamente, el modelo aquí propuesto ha sido determinado para una población de virus de inmunodeficiencia humana (VIH) *in vivo*. En función de ello, se han considerado dos especies para la formación del modelo: por una parte, el propio virus; por la otra, el sistema inmunológico del paciente. Los individuos de la primera especie, mutan, debido a su forma de reproducción, lo que provoca una permanente evolución del sistema y de la función de optimización implícita, vista como su capacidad para escapar del sistema inmunológico.

En los algoritmos genéticos, mediante una codificación, se representan las configuraciones como cromosomas; una forma de hacer esta codificación es indirecta, según lo plantea Kauffman utilizando unidades reguladoras, basándose en que éstas se comportan, *grosso modo*, como la información genética de un organismo vivo.

En el modelo se utilizó codificación indirecta mediante autómatas de Kauffman, que si bien no se asemeja mucho a la codificación "natural" de los virus, comparte con ella ciertas características esenciales, como la de ser un procesador no local de información genética, y la existencia de una gran degeneración (como se explica en el tercer capítulo), es decir, existen muchas posibles secuencias genéticas equivalentes al mismo organismo vivo.

1. Antecedentes biológicos

Introducción

Los seres vivos son aquellos que pueden nutrirse, crecer y reproducirse,⁶ ya sea que estén formados por una o varias células. Así pues, como seres vivos, las células son capaces de nutrirse, crecer y reproducirse.

La reproducción de las células se realiza de manera directa, a partir de una célula madre, mediante el crecimiento de todos sus componentes, sin que ésta pierda su integridad. Para llevar a cabo esta actividad, las células constan de una maquinaria sintética y metabólica, que, a partir de ácido desoxirribonucleico (ADN), produce ácido ribonucleico (ARN) mensajero y de transferencia, lo cual, tras una serie de procesos, genera un aumento de los componentes de la célula para dar origen a una nueva.

La información genética de todo organismo viviente se encuentra codificada en los citados ácidos nucleicos, que son cadenas de combinaciones de 20 aminoácidos y una señal de terminación. Durante la reproducción de las células, en la síntesis de proteínas, la información genética fluye del ADN hacia el ARN para formar proteínas.

Los virus y su reproducción

Los virus son considerados los organismos vivos más pequeños. Un virus es un paquete de información genética, constituido por un solo ácido nucleico (ADN o ARN) que carece de metabolismo y para cuya reproducción requiere de la maquinaria genética de una célula infectada.

El ácido viral puede estar cubierto por una proteína, que lo envuelve y estabiliza. Al entrar en contacto con una célula susceptible al virus, su envoltura facilita la unión y penetración del ácido.

Tras la penetración del ácido viral que contiene todo el material genético para programar la célula huésped, ésta reproduce los componentes de un virus en forma independiente. Por su parte, estos componentes se ensamblan para formar partículas víricas maduras, que escapan de la célula. En ocasiones, la célula —cuando ha sintetizado una gran cantidad de partículas virales— sufre una lisis, es decir, su membrana se rompe y deja

⁶ A excepción de los virus, que, aunque considerados seres vivos, no pueden realizar todas estas tareas

escapar los virus que hayan quedado en su interior, destruyéndose así la célula.

Los virus formados por ARN son llamados retrovirus. Éstos con la ayuda de una enzima llama transcriptasa inversa (*reverse transcriptase*), al penetrar en una célula, llevan a cabo un proceso de transcripción inversa para producir ADN viral (a partir de su ARN), que insertan en el ADN de la célula hospedera. La transcripción inversa tiende a ser poco precisa y suele producir algunos errores que repercuten en mutaciones.

La respuesta inmune

Un organismo animal (incluido el humano) tiene básicamente dos mecanismos de protección: los no específicos, también llamados resistencia natural o innata, y los específicos, representados por la respuesta inmune.

La respuesta inmune es una serie de mecanismos cuyo objetivo general es defender al organismo de cualquier invasor específico. Puede ser considerada como un proceso en el que participan linfocitos T, linfocitos B, macrófagos, leucocitos PMN, y anticuerpos.

Es importante mencionar que la respuesta inmune se desencadena por la detección de un antígeno. Un antígeno es una sustancia extraña, que, introducida en un individuo, provoca la formación de anticuerpos que reaccionarán específicamente contra este determinado invasor [D. P. Stites]. Para que una molécula sea considerada como un antígeno, son importantes su exogenicidad (su carácter de extraño), tamaño molecular, complejidad química, estructura secundaria y terciaria, la constitución genética del individuo y el modo de administración del antígeno, así como su capacidad para unirse con otras moléculas para formar otras con propiedades antigénicas mayores.

Lo que en realidad implica la respuesta inmune son los determinantes antigénicos, constituidos por áreas que determinan las reacciones antígeno-anticuerpo. Aunque las grandes proteínas expresan múltiples posibles determinantes, un individuo elabora anticuerpos sólo contra un pequeño subgrupo del total. Podemos decir que los determinantes antigénicos son áreas expuestas o prominentes en la superficie de las moléculas; contra estas zonas van dirigidas las respuestas inmunes. En general, el número de determinantes antigénicos sobre una molécula está en relación directa con el tamaño de la misma. Es probable encontrar determinantes antigénicos

idénticos en cierto número de moléculas diferentes, de modo que los anticuerpos que corresponden a uno de los antígenos, pueden reaccionar con antígenos cuyo origen parece muy diferente. Se habla entonces de *reacción cruzada*.

Los macrófagos "inician" la respuesta inmune al capturar al ente extraño, al que llamaremos antígeno, a continuación, en combinación con los leucocitos PMN, lo procesan, obteniendo de él su identificación (epitopo), mientras que el resto se degrada por digestión fagocítica.

Los linfocitos T —células T auxiliares— reciben de los macrófagos el epitopo de reconocimiento del antígeno a través del complejo antígeno-anticuerpo receptor /CD3, a partir de lo cual estimulan la producción de células T asesinas —encargadas de la destrucción de células infectadas por dicho antígeno—, así como la de linfocitos B.

Cuando un macrófago, en compañía de una célula T auxiliar, fagocita un antígeno, la temperatura corporal aumenta para estimular la actividad de las células del sistema inmunológico. Por su parte, los linfocitos B (células B) forman anticuerpos contra el antígeno identificado por las células T auxiliares. Finalmente, los anticuerpos se adhieren a los antígenos, inactivándolos y marcándolos para ser eliminados por los macrófagos.[◊]

El virus de la inmunodeficiencia humana (VIH)

El VIH es un retrovirus, lo que implica que debe llevar a cabo el proceso de transcripción inversa, a fin de introducir su información en la maquinaria genética de la célula hospedera. En particular, el VIH tiene alrededor de 10^{-4} errores por base y por ciclo [M. A. Nowak, 1992], lo que implica que la transcripción inversa genera una mutación exitosa en el 60% de las infecciones (alrededor de una mutación exitosa por día). Este dato nos habla de lo altamente mutable que es el virus.

En general, un virus es un antígeno; las pequeñas áreas de su superficie, llamadas determinantes antígenos o epítomos, son reconocidos por los anticuerpos para su destrucción. Ahora bien, si estos epítomos cambian como resultado de las mutaciones, tenemos un nuevo determinante antigénico, en

[◊] En su mayoría, las células T son maduradas por el timo; mientras que las células B se generan en el hígado, durante la etapa fetal, y, superada ésta, en la médula ósea. La producción de cada clon de estas células depende también del complejo mayor de histocompatibilidad (MCH). Los macrófagos, cuya vida promedio es de un mes, se producen también a partir de la médula ósea.

consecuencia, es posible que no se cuente con la infraestructura necesaria para contrarrestarlo. De hecho, si tomamos en consideración que una variedad (cepa) de virus puede producirse cada día, mientras que el desarrollo y acción de una respuesta inmune, para un nuevo determinante antigénico toma alrededor de tres semanas, comienza a ser obvia la dificultad a la que se enfrenta el sistema inmunológico, para que un individuo pueda eliminar toda partícula viral de VIH.

El hecho de que un paciente contraiga el VIH, no implica que deba morir; en general, puede ser portador del virus sin síntomas de enfermedad, sanar, o enfermar. Esto depende de su sistema inmunológico, ya que a pesar de que debe eliminar a un virus altamente mutable, hay partes del VIH que no pueden cambiar, a estas partes las llamaremos epítomos conservados.

La dinámica que se presenta depende de si el sistema inmunológico elabora una respuesta en contra de los epítomos conservados, o contra los altamente mutables. En caso de tomar como determinante antigénico la parte que nunca cambia, el VIH es visto como cualquier otro virus, con una sola cepa contra la cual se tienen linfocitos B, T y anticuerpo; esto lleva a eliminarlo en un plazo de tiempo lo suficientemente corto como para que no cause mayor daño al paciente.

Pero si el determinante antigénico es un epítomo altamente mutable, el sistema inmunológico parecerá enfrentarse a una enfermedad distinta cada día, lo que demanda una gran cantidad de recursos para desarrollar una respuesta específica contra cada una de ellas. Aunque este panorama es similar al de la influenza (gripa) producida también por un retrovirus, hay que destacar que a diferencia de ésta, que infecta tejidos de vías respiratorias, el VIH infecta células del sistema inmunológico, ya que presenta tropismo relativo por linfocitos que tienen en su cubierta antígenos T4, en particular los linfocitos T auxiliares y T superiores. Además, el VIH se reproduce con mayor rapidez si las células T se encuentran estimuladas por un antígeno, e incluso su tasa de reproducción es mayor que la de las T.

En la “batalla” entre el sistema inmunológico y el VIH, éste último tiene una ventaja por su no-especificidad: mientras los anticuerpos son necesariamente específicos para un epítomo, los virus atacan a cualquier célula CD4 del sistema.

2. Modelo matemático

En este capítulo se plantea un sistema de ecuaciones diferenciales, por medio de las cuales se pretende modelar el cambio en las poblaciones de virus y células con las que cohabitan. M.A. Nowak [*et al.*, 1992] propone el siguiente sistema de ecuaciones para los incrementos de poblaciones virales y la respuesta inmune específica asociada:

$$\frac{dv_i}{dt} = v_i(r - px_i) \quad (1)$$

$$\frac{dx_i}{dt} = kv_i, \quad i = 1, \dots, n \quad (2)$$

En estas ecuaciones v_i representa la cantidad de virus de la cepa i ; x_i las células específicas para su repuesta inmune; r es la tasa de crecimiento de los virus; px_i representa la disminución en la población de virus debida a la respuesta inmune específica; p es la probabilidad de que un virus sea eliminado por la respuesta inmune, la cual es proporcional a v_i .

Interpretados, los factores anteriores indican que la población de virus de la cepa i aumenta con una tasa r , lo que significa que de un virus se producen r igual a él. La disminución de su población está dada por encuentros entre virus y células con probabilidad proporcional a p .

La segunda ecuación muestra que el crecimiento de la población de células inmunológicas es proporcional a la cantidad de virus, ya que cuando el sistema inmunológico detecta una población de virus, su respuesta es la producción de una cantidad proporcional de células de respuesta inmune; n es el número de cepas de virus y de respuestas asociadas, el cual se incrementa según se produzcan nuevas mutaciones del virus.

Si se tiene un modelo más complejo, de manera que se incluya la muerte de células inmunológicas, las ecuaciones son las siguientes:

$$\frac{dv_i}{dt} = v_i(r - px_i) \quad (3)$$

$$\frac{dx_i}{dt} = kv_i - uvx_i, \quad i = 1, \dots, n \quad (4)$$

Para este nuevo sistema, el término uvx_i representa la muerte —debida a infección viral— de células inmunológicas; mientras que v es la cantidad total de virus.

El sistema inmunológico puede controlar la cepa i , tanto como las diferentes cepas, si $r - px_i < 0$ ($v_i < 0$), lo cual equivale a acotar la tasa de crecimiento viral y la de su muerte: $n < nc = px_i r$. Esto es, si el número de cepas permanece por debajo de la razón de muerte y reproducción de virus, que se ve reflejado cuando la tasa de mortalidad es mayor que la de reproducción.

Para lograr la extinción de todos los virus, el sistema inmunológico debe enfrentar un tipo de “latencia”, cuando se encuentran infectadas algunas células que no pueden ser atacadas por él, como las del sistema nervioso, o las de las corneas; esto incrementa la habilidad de mantener una población viral permanente. Si representamos estos depósitos de virus como w_i , obtenemos el sistema modificado de la siguiente forma:

$$\frac{dv_i}{dt} = v_i(r - px_i - \alpha) + \beta w_i \quad (5)$$

$$\frac{dx_i}{dt} = kv_i - uvx_i \quad (6)$$

$$\frac{dw_i}{dt} = \alpha v_i + kv_i - uvx_i - \gamma w_i \quad i = 1, \dots, n \quad (7)$$

Aquí, w_i indica el número de virus en los depósitos, βw_i es la tasa de reactivación de virus desde los depósitos, αv_i es la tasa de producción de depósitos (virus que salen de la población “activa”), mientras que γw_i representa la tasa de muerte de células depósito.

Por otro lado, si se toma en cuenta que puede presentarse una respuesta inmune inespecífica, las ecuaciones deben modificarse de la siguiente manera:

$$\frac{dv_i}{dt} = v_i(r - sz - px_i) \quad (8)$$

$$\frac{dx_i}{dt} = kv_i - uvx_i \quad (9)$$

$$\frac{dz}{dt} = k'v - uvz \quad i = 1, \dots, n \quad (10)$$

Para este sistema, z representa la respuesta inmune no específica, es decir, lo mismo reduce a un tipo de virus que a otro en una cantidad k proporcional al tamaño de su población, mientras que se genera k' veces del total de virus, pero también disminuyen de forma proporcional al total de virus, sin importar su tipo (v).

El sistema de ecuaciones diferenciales {(8), (9), (10)}, refleja la variación de las distintas poblaciones que intervienen en el caso de estudio tomando algunas consideraciones.

Del sistema {(8), (9) y (10)} para los valores de los parámetros: $r = 1$, $p = 2$, $s = 0.93$, $u = 3$, $k = k' = 3$, y cambiando sólo la semilla para generar los números aleatorios, se observan comportamientos distintos de las poblaciones, como se puede apreciar en las siguientes gráficas reproducidas del trabajo de Nowak [et al., 1995]:

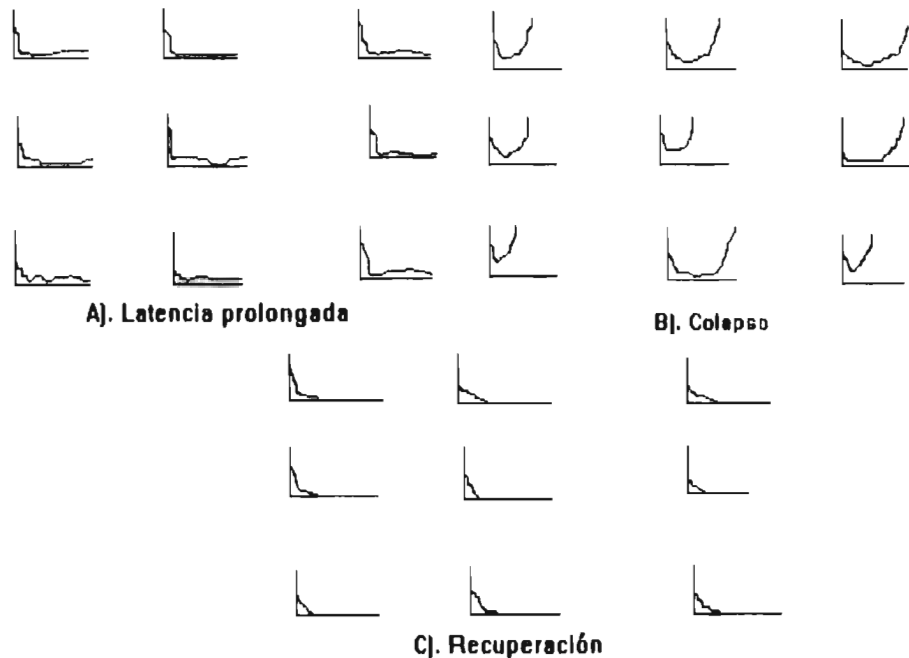


Fig. 2.1 Gráficas obtenidas de [14] simulaciones del sistema (8),(9),(10) con $r=1$, $p=2$, $s=0.93$, $u=3$, $k=k'=3$. En cada gráfica el eje horizontal indica el tiempo desde la infección, mientras que el vertical es el tamaño de la población de virus.

La gráfica muestra en el eje y el tamaño de la población de virus; en el x , el tiempo. Se pueden observar curvas asintóticas a x , correspondientes a casos de latencia (A); otras referentes al colapso del sistema (B), y también de recuperación (C).

3. Modelo genético

Introducción

Un algoritmo genético (AG) es un método de búsqueda de óptimos, que toma como ejemplo el proceso de reproducción de individuos y selección natural. Los individuos, sean de la especie que sean, deben pasar una serie de pruebas (v.g. amenaza de los depredadores, enfermedades y escasez de alimentos), antes de alcanzar su madurez y poder reproducirse. Al hacer esto último, heredan características a sus descendientes.

Los AG llevan a cabo la tarea de optimización, considerando —por una parte— cada elemento del espacio de búsqueda como un individuo, y —por otra— una cierta función de optimización como el medio ambiente. Estos individuos están codificados por cromosomas, que representan todas las variables del problema. El cromosoma es una lista de símbolos, que, mediante la acción de un intérprete, se traduce en una lista de variables, las cuales representan a un individuo.

La población del modelo está constituida por un cierto número de cromosomas. Su evolución resulta de la aplicación de “operadores genéticos”: reproducción, entrecruzamiento y mutación, principalmente. El operador de mutación consiste en el cambio aleatorio en el valor de uno (o varios) de los genes. Mediante entrecruzamiento se da una recombinación de los genes de dos cromosomas “padres” para formar dos nuevos. El operador de reproducción evalúa la capacidad de los individuos para librar las adversidades (optimizar la función). La “habilidad” de cada individuo dependerá del valor de la función que represente. Al aplicar el operador de reproducción se generan copias de cada individuo en una cantidad proporcional a su habilidad. La aplicación de los operadores da lugar a una nueva población. El cromosoma más apto de entre todos los posibles cromosomas representa al óptimo dentro del espacio de soluciones. Este no necesariamente se encuentra en la población “actual”.

Los AG son un método de búsqueda independiente del problema. La obtención de nuevos individuos, mediante el uso de operadores genéticos, se puede realizar sin el menor conocimiento del problema. Toda la información relacionada con el problema específico se refleja en la codificación de las variables para interpretar los cromosomas, y el planteamiento de la función de evaluación para determinar la aptitud de cada cromosoma.

Existe una gran variedad de métodos de optimización. La eficiencia de cualquier método de aplicación dependerá de la naturaleza de la búsqueda; no existe un método de optimización que sea el mejor para todos los problemas. Los AG han demostrado ser eficientes en la modelación de sistemas complejos, o en problemas de optimización relacionados con topografías rugosas, en donde los otros métodos no son fácilmente aplicables.

En general, un algoritmo genético está caracterizado por: su codificación, la técnica de arranque utilizada para producir la población inicial, la función de evaluación, los operadores genéticos (que permiten modificar las poblaciones), la forma de reemplazar la(s) población(es), el criterio o criterios utilizados para concluir la aplicación de los operadores genéticos, los parámetros que determinan la probabilidad de aplicación de cada uno de los operadores genéticos, y el número de elementos de la población.

Codificación (genotipos y fenotipos)

En forma natural, la función de aptitud se aplica directamente sobre los individuos físicos (fenotipos). Sin embargo, al elaborar un modelo no se puede trabajar con dichos individuos, por lo que se representan sus características mediante una codificación como una cadena de caracteres. La codificación es un paso importante en los algoritmos genéticos, pues permite almacenar información del espacio de soluciones dentro los cromosomas, además de que —en conjunción con la función de evaluación— expresa la dinámica del sistema. Por ejemplo: sea G el espacio de genotipos, constituido por el conjunto de cadenas codificadas (cromosomas), y Q el conjunto de fenotipos o individuos “reales” en el espacio de soluciones. Un fenotipo puede ser el género (sexo) del individuo; mientras que sus genotipos serán la parte correspondiente al par 23 de su cromosoma (XY y XX) (ver la figura 3.1).

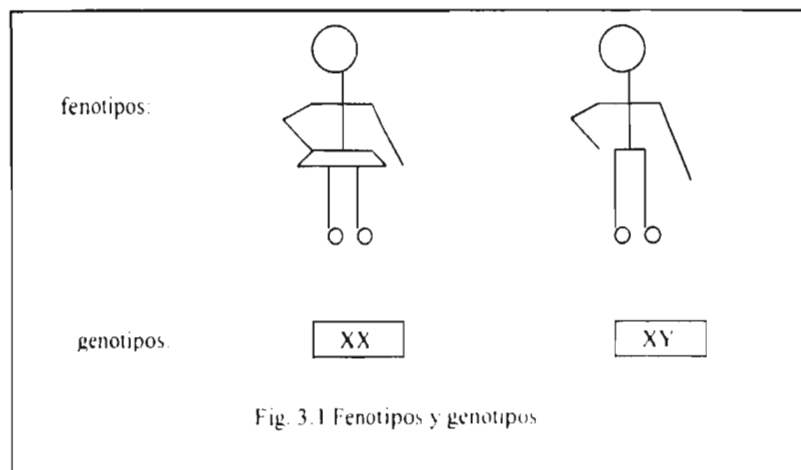


Fig. 3.1 Fenotipos y genotipos

La codificación se puede hacer de distintas formas. En el método llamado de codificación indirecta, las variables del problema se representan en los genes del cromosoma mediante el uso de un intérprete ($\Phi: G \rightarrow Q$), que traduce los valores de un cromosoma en valores de las variables del problema. En la codificación directa, en cambio, no se requiere del intérprete que relacione genotipos y fenotipos, pues un cromosoma representa un y solo un individuo en el espacio de soluciones, y viceversa.

En la codificación indirecta, el intérprete Φ no necesariamente tiene correspondencia de uno a uno; en general, tampoco es local, no se tiene definido cada gen (o localidad en el cromosoma) como una característica en el fenotipo. Como, según ya se ha indicado, la correspondencia no es uno a uno, aparece el concepto de sinónimos (*i.e.* existen varios genotipos distintos que codifican el mismo fenotipo). Así pues, los genotipos g_1 y g_2 son sinónimos ($g_1 \sim g_2$) si y sólo si $\Phi(g_1) = \Phi(g_2)$.

La función de aptitud F que asocia un valor en \mathbf{R}^+ a cada uno de los fenotipos representados por cromosomas, es una relación $F: Q \rightarrow \mathbf{R}^+$. La aptitud o habilidad inducida $V = F \circ \Phi$ relaciona la codificación con la función de aptitud F , asignando un valor real positivo a cada cromosoma ($V: G \rightarrow \mathbf{R}^+$).

La relación entre la codificación y la función de aptitud se puede observar en la figura 3.2. Los genotipos (elementos de G) son asociados con los fenotipos (elementos de Q) mediante la codificación Φ . Estos fenotipos son evaluados por la función de aptitud F para obtener un valor real positivo. Para la codificación directa tanto Φ como F son biyectivas; mientras, que la codificación indirecta permite que Φ pueda ser suprayectiva, sin ser inyectiva (*i.e.* permite la aparición de “sinónimos”).

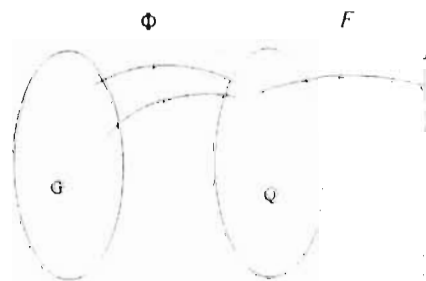


Fig. 3.2 Correspondencia de los genotipos a los reales

La función de aptitud NK y su topografía (fitness NK)

La función de aptitud $F: Q \rightarrow R^+$ asocia un valor numérico que determina la habilidad de cada individuo. Generalmente, en la modelación de sistemas complejos es difícil establecer la función de aptitud, debido a las interacciones no locales y no lineales entre los elementos del cromosoma. Para modelar este tipo de sistemas, se utiliza la función NK. Ésta consiste en asignar una contribución aleatoria a cada posición del cromosoma de longitud N , con base en su valor y el de sus K vecinos relacionados (no necesariamente vecinos físicos). Esto se ilustra en el siguiente ejemplo:

$K=2$, indica que cada gen depende de otros dos, y $N=5$ es la longitud del cromosoma.

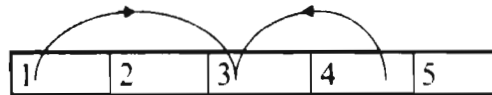


fig. 3.3 Cromosoma con longitud $N = 5$ y número de conexiones $K = 2$

De la figura se observa que la contribución de la tercera posición al valor de aptitud de la cadena, depende de los alelos (valores) en las posiciones 1, 3 y 4. Para evaluar esta contribución se usa una tabla de ocho números generados aleatoriamente para cada posición (ver tabla 3.1). En general, se construye una tabla de n^{K+1} renglones, en donde n es el tamaño del alfabeto usado para los alelos. Cada renglón representa una de las posibles combinaciones de longitud K de n caracteres. Para nuestro ejemplo, basados en un alfabeto binario, obtendríamos la tabla 3.1, en donde

Pos.1	Pos.3	Pos.4	Aportación
0	0	0	0.427
0	0	1	0.781
0	1	0	0.369
0	1	1	0.015
1	0	0	0.492
1	0	1	0.276
1	1	0	0.644
1	1	1	0.305

Tab. 3.1 Ejemplo de la aportación de la posición No 3 a la función de optimización NK, con $N=5$, y $K=2$

las primeras tres columnas de la tabla indican las posibles combinaciones de valores en la posición número tres y sus vecinas y la cuarta columna indica su aportación (un número aleatorio) a la función de optimización.

Así por ejemplo, si tenemos los cromosomas:

a)

0	1	0	0	1
---	---	---	---	---

 b)

1	1	0	0	1
---	---	---	---	---

 c)

0	1	0	1	1
---	---	---	---	---

 y d)

0	1	1	0	1
---	---	---	---	---

entonces, la aportación en cada caso se obtiene como sigue: Para el cromosoma (a), las posiciones 1, 3 y 4 valen 0, por lo que contribución de la tercera posición es: 0.427 (renglón 000 de la tabla), para el cromosoma (b), la aportación es de 0.492 (renglón 100), para el (c), la aportación es de 0.781, y para el (d) es 0.369.

Aunque en los tres primeros casos el valor almacenado en la tercera posición es invariable, su aportación a la evaluación del cromosoma cambia. La aportación de una posición no sólo depende del valor que ésta almacena, sino que también interviene el de las posiciones relacionadas (1 y 4 para el ejemplo). El cambio de valor en una posición altera su aportación, pero también la de las referentes a todas aquellas posiciones de las cuales es una posición relacionada. En el ejemplo, el valor de la posición uno repercute en su aportación, lo mismo que en la de la posición tres (cromosomas (a) y (b)).

El valor de la función de optimización NK para el cromosoma se calcula mediante la suma de las aportaciones de las N posiciones.

La gráfica que se obtiene representando a los fenotipos en el eje horizontal y su valor asociado por la función de optimización en el eje vertical, a la que llamaremos *topología de la función* de aptitud, nos permite identificar el *grado de frustración*[‡] existente en el problema.

[‡] Entendamos como frustración aquella situación en la que el cambio de valor en una posición específica influye en la aportación de otras al resultado de la evaluación del cromosoma. Pues si bien un lugar x tiene su valor óptimo, al modificar otro y, x deja de ser óptimo.

En el modelo NK, la frustración y por lo tanto, la rugosidad de la función de aptitud (topología), crecen con el valor de K. En un cromosoma de longitud N, $K = 0$ representa a un problema sin interacciones no lineales, cuyo paisaje tiene un solo óptimo y no existe frustración. Si el número de interacciones no lineales es $K = N-1$, tenemos el más alto nivel de decepción o frustración y un paisaje lleno de picos. Por lo tanto, dos cromosomas separados por una distancia de Hamming mínima generalmente representan soluciones muy diferentes. El resultado de evaluar la función de optimización, para los dos cromosomas vecinos, genera dos resultados aleatorios estadísticamente independientes. Para un experimentador, es como si la función de evaluación generara valores aleatorios para los diferentes fenotipos.

Cuando conocemos la relación entre las diferentes variables podemos determinar una función de optimización analítica. Pero para sistemas adaptativos, no siempre es posible (ni fácil) determinar una función matemática, ya que la relación entre las diferentes variables es cambiante. Es en este escenario donde los AGs resultan eficientes, porque pueden permitir a la aptitud de los cromosomas evolucionar a la vez que el sistema lo hace (se tiene una función implícita en los cromosomas).

Algoritmo de Kauffman

Una manera de llevar a cabo la codificación indirecta es mediante el modelo propuesto por Kauffman, basado en autómatas celulares. Un cromosoma es una lista de genes, vistos como unidades reguladoras o reglas, con un número k de interacciones o referencias a distintos sitios a lo largo del cromosoma. La regla indica como se relacionan estos k genes.

Para codificaciones sobre un alfabeto binario, la regla es una función booleana. La regla se selecciona de entre 2^x , en donde $x = 2^k$, reglas booleanas posibles, con base en su probabilidad de generar un *punto fijo*. Un punto fijo es un fenotipo codificado por un cromosoma, que al evaluar cada una de sus reglas con sus respectivas conexiones, permanece sin cambios. Por ejemplo para un cromosoma de longitud 3 con $k = 2$ y el esquema de conexiones y reglas dado por la tabla 3.2, podemos ver que un punto fijo es el 000.

reglas	AND	AND	AND
conexión 1	2	1	1
conexión 2	3	3	2
	gen 1	gen 2	gen 3

Tab. 3.2 Ejemplo de autómeta celular con reglas booleanas y dos conexiones.

En efecto, como el gen 1 cambia de acuerdo al resultado de la operación AND entre el gen 2 y 3 su nuevo valor será 0, el nuevo valor del gen 2, se calcula a partir del AND entre 1 y 3 y por último el nuevo valor del gen 3, será el resultado del AND entre 1 y 2. Vemos entonces, que el resultado de la aplicación de las reglas es el cromosoma original (000), por lo tanto tenemos un punto fijo. Sin embargo, la cadena 101 no es un punto fijo, ya que el nuevo cromosoma sería 010, como puede verificarse fácilmente (ver tabla 3.3).

cromosoma original	0	0	0	1	0	1
	AND	AND	AND	AND	AND	AND
autómata						
cromosoma resultante	0	0	0	0	1	0

Tab. 3.3 Ejemplo de la evolución de cromosomas por el autómeta celular de la tabla 3.2.

El modelo de Kauffman intenta representar la regulación de la expresividad de los genes en una célula. Así, las interacciones entre autómetas representan la forma en que un gen puede estimular o inhibir a otro, como sucede en los cromosomas reales. Al llegar el autómeta a encontrar un punto fijo, se considera que la "célula" llega a un patrón estable de actividad.

Generalmente, el valor de k usado para los autómetas celulares es 2, ya que para $k > 2$, la probabilidad de llegar a puntos fijos es muy baja, mientras que para $k < 2$ no se encuentra una dinámica interesante. Con este valor de k , se ha demostrado empíricamente [Kauffman, 1989] que cada cadena reguladora admite una cantidad del orden de \sqrt{N} puntos fijos posibles. Para células humanas, esto significa que los aproximadamente 10^4 genes (unidades reguladoras) conducen a unos 10^2 distintos tipos de células (puntos fijos), que difieren entre sí, por el estado de actividad de sus genes.

Estructura de los Algoritmos Genéticos

Hasta el momento se han visto partes de un AG aisladas, pero el algoritmo en sí, es una secuencia de las siguientes acciones:

1. Génesis (generación de la población inicial).
2. Evaluación de aptitud de los elementos de la población.
3. Selección de individuos.
4. Reproducción.
5. Entrecruzamiento.
6. Mutación.
7. Reemplazo de la población anterior.

Tales acciones se definen como sigue:

1. *Génesis*: Es la generación de la población inicial, la cual puede ser aleatoriamente, o como resultado de experimentos anteriores u otros modelos.

2. *Evaluación*: Mediante esta acción se asigna un valor de R^+ a cada individuo. Este valor representa su habilidad o aptitud.

3. *Selección*: En este paso se seleccionan los individuos más aptos, o mejor adaptados, para lo cual se utilizan principalmente los métodos de la ruleta y el torneo.

En la ruleta, cada individuo está representado por un área proporcional a su aptitud en una rueda, la porción de la rueda correspondiente a cada individuo se calcula como el cociente de su aptitud entre la aptitud de la población (suma de todas las aptitudes de todos los individuos). La rueda se gira y al detenerse un "apuntador" señala al individuo que pasará a la siguiente generación (el individuo correspondiente a la porción del círculo señalada), de esta forma la probabilidad de que un individuo sea seleccionado es proporcional a su aptitud.

Por ejemplo, supongamos que tenemos una población de tres individuos, A, B y C cuyas aptitudes son 6, 10 y 4 respectivamente. Entonces, el área correspondiente a A es de $6/20$ del área total $10/20$ la de B y $4/20$ la de C, como se muestra en la figura 3.6.

Individuo	Aptitud
A	6
B	10
C	4

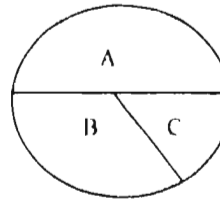


Fig. 3.6 Rueda de ruleta para tres individuos

En el torneo se escoge un conjunto de individuos, haciéndose un comparación de las aptitudes (fitness) de estos. El de mayor aptitud se considera ganador del torneo y pasará a la generación siguiente.

4. *Reproducción:* Aquí se generan copias de los individuos seleccionados para tener una población de N elementos en la siguiente generación. Para esto se deben dar N giros a la ruleta u organizar N torneos.

5. *Entrecruzamiento:* Se seleccionan al azar un par de individuos (padres) y una posición en el cromosoma (punto de cruce). La posición indica el punto en el que ambos individuos serán divididos, para después recombinar las partes de ambos y generar nuevos individuos en la población (hijos). Esto se ilustra en el siguiente ejemplo:

Supóngase que han sido seleccionados los individuos 01010 y 00101, un punto de cruce igual a 3. La siguiente generación se obtendría entonces como se muestra en la tabla 3.4.

0	1	0	1	0	Padre 1
0	0	1	0	1	Padre 2
0	1	0	0	1	Hijo 1
0	0	1	1	0	Hijo 2

Tab. 3.4 Ejemplo de entrecruzamiento.

6. *Mutación*: Consiste en cambiar el valor de un gen por uno distinto al que tenía. En una codificación binaria, esto equivale simplemente a reemplazar un cero por un uno o viceversa.

7. *Reemplazo*: Los operadores de entrecruzamiento y mutación, se aplican a las copias de los elementos seleccionados, con una cierta probabilidad, después de lo cual se sustituye la población anterior por la generada.

Los pasos del 2 al 7 se repiten una y otra vez, hasta que cumplan condiciones de paro determinadas de antemano.

El modelo para el VIH (Algoritmo Genético para el VIH *in vivo*)

Deseamos modelar el comportamiento poblacional del *VIH* dentro de un organismo vivo. En este sistema tenemos un par de especies, el virus y células T^{*}, que co-evolucionan. Ambas especies son competitivas y la función de aptitud estará dada por la interacción entre ambas especies. Como no es posible determinar una función de aptitud que represente esta interacción, usaremos un modelo adaptativo.

Codificación:

La interacción entre una población de virus y las células T, se da mediante la identificación del virus por parte de las células. Las posibles zonas de reconocimiento del virus, se encuentran en su proteína *env* (envoltura). En este caso, el sistema inmunológico tiene problemas para reconocer al virus debido a su alta mutabilidad, sin embargo, la funcionalidad de esta proteína implica ciertas restricciones sobre la cadena de aminoácidos, que constituyen su estructura primaria. Las restricciones funcionales son debidas a la estructura secundaria y terciaria de la envoltura del virus y relacionan aminoácidos que no son necesariamente vecinos cercanos.

Las mutaciones pueden suceder en cualquier parte de la estructura del virus, si la mutación produce una violación a las restricciones funcionales, el virus morirá. De allí que existan mutaciones exitosas y mutaciones no exitosas.

Para modelar las interacciones no locales entre aminoácidos usaremos los autómatas de Kauffman antes mencionados.

* Usamos a las células T como representantes del sistema inmunológico

El propósito de este intérprete, será determinar si una mutación en el cromosoma es viable o no. Una mutación viable permitirá al cromosoma (autómata) llegar a un punto fijo, de esta manera, un punto fijo en el autómata de Kauffman, representará para nosotros una mutación que no viola las restricciones funcionales entre la estructura del virus.

El algoritmo se basa en secuencias de 16 “letras” (cromosomas) y la codificación indirecta se aplica con autómatas celulares $K=2$, es decir, cada letra del cromosoma está compuesto por una regla booleana de la 0 a la 15 y un par de conexiones. Esta forma tiene un total de 4,096 diferentes unidades reguladoras posibles.

Representamos a un epítipo (determinante antigénico) como una unidad reguladora (regla y par de conexiones) en el cromosoma del modelo. La envoltura de un virus se modelará por cromosomas de longitud 16. Este tipo de cromosomas es usado para representar tanto a los virus como a las células infectadas. A los virus se asocia una virulencia, representada por un número aleatorio entre 0 y 1. Los cromosomas que representan la respuesta inmune son de longitud 1 equivalente a un epítipo.

Para el modelo, se omiten las conexiones de epítipos con partes del virus que no pertenecen a la envoltura.

Cada virus está representado por tres vectores de longitud 16 ($W1$, $W2$ y R) y un número real VV . Los primeros dos vectores indican el par de vecinos de cada gen, mientras que R muestra la regla booleana con la que se evalúa. como se ilustra en la tabla 3.5.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$W1$	3	7	5	0	8	12	13	6	9	7	2	14	11	3	1	2
$W2$	6	0	5	14	9	7	15	4	6	5	0	4	12	11	8	2
R	10	9	15	12	9	4	6	11	5	12	13	3	8	10	14	4
VV	0.21															

Tab. 3.5 Ejemplo del cromosoma de un virus en el modelo.

Por ejemplo, la posición número 7 tiene como vecinos a los genes 6 y 4, y se evalúa mediante la regla booleana número 11 (tabla de reglas booleanas

en el Apéndice B). El número real $VV \in [0,1]$ indica la virulencia, así que, este virus infecta al 21% de las células que encuentra.

Las células infectadas están representadas por tres vectores y un número entero que indica la edad de la célula. Los vectores son de longitud 16, con la misma función y sobre el mismo alfabeto que los virus.

Las células del sistema inmunológico, llamadas células T, constan de tres vectores de longitud 1, que representan un gen de los virus o las células infectadas.

Técnica de arranque:

Se genera una cantidad aleatoria (en promedio, la décima parte de la población total) de individuos del sistema inmune, de longitud 1, y una cantidad, también aleatoria, de virus de longitud 16. Los individuos están formados por reglas booleanas y pares de conexiones, generadas también aleatoriamente. Los cromosomas de longitud 16, deben conducir a un punto fijo, en menos de 30 iteraciones, para ser tomadas como virus, en caso contrario se desechan.

Función de evaluación:

En realidad se cuenta con dos funciones de evaluación implícitas y diferentes, una para los virus y otra para las células T. La aptitud de los virus está relacionada con su capacidad de escapar al sistema inmunológico y su virulencia y la aptitud de las células T, se da según su capacidad de reconocer a los virus, sin ser infectadas.

Operadores genéticos:

El único operador genético utilizado en este modelo, es la mutación. Deseamos modelar el comportamiento del virus, usando como referencia su alta mutabilidad debida a la transcripción inversa. El entrecruzamiento no se utiliza, pues ni las células ni los virus, necesitan una pareja para su reproducción. Los virus se reproducen mediante la infección de células que producen un número α de nuevos virus cada una.

Técnica de reemplazo:

Se cuenta con una población de virus, una de células infectadas y una para las células T. Las nuevas poblaciones se generan de la siguiente forma: Se producen células infectadas en cantidad proporcional a los virus, de manera que cada célula da lugar a α virus, las células de la respuesta inmune, se incrementan en k veces la población de virus. La población de células infectadas decrece por efecto de la lisis, en cantidad proporcional a los virus. Estos disminuyen su número, debido al sistema inmunológico y finalmente el número de células del sistema inmunológico decrementa proporcionalmente a la cantidad de virus.

Esta dinámica, es la observada en una población de cualquier virus *in vivo*.

Cuando se han completado estos pasos, se integran las nuevas poblaciones con los elementos sobrevivientes de las anteriores y los que se generaron en el proceso.

Criterios de terminación:

El momento para dejar de manipular las poblaciones ocurre cuando se da alguna de las siguientes situaciones:

- a) Colapso del sistema (muerte del paciente), que se presenta al alcanzar un número máximo de células infectadas (800).
- b) Recuperación total, es decir, cuando las poblaciones de virus y de células infectadas son cero.
- c) Cuando se rebasa un número de iteraciones (450).

Ecuaciones Iterativas

El comportamiento de las tres poblaciones (virus, células infectadas, y células T) que coexisten en el sistema, se puede expresar con tres ecuaciones iterativas, tomadas del problema real y del Algoritmo Genético.

La cantidad de virus de una cepa j en el sistema en un cierto tiempo $n+1$, es la del tiempo anterior mas una cantidad proporcional a la del virus

que se generará en las células infectadas por esta cepa, menos la correspondiente a los que "murieron" a causa del sistema inmunológico.

La cantidad de células infectadas por la cepa j en el tiempo $n+1$ es: la parte que sobrevivió al sistema inmunológico y a la lisis (δ, ϵ), mas una cantidad proporcional a la de los virus con el epítipo j , que no mutaron, mas una parte proporcional de los virus que no perdieron al epítipo j por una mutación puntual, mas una parte proporcional de los que lo perdieron y lo reconstruyeron, mas una parte proporcional de los que no lo tenían y lo generaron por mutación.

Si μ es la probabilidad de que un epítipo mute hacia cualquier otro, entonces el número de epítipos de la cepa j que no mutaron es $(1-\mu) V_n^j$. Llamemos r al número total de posibles epítipos, así que la probabilidad de seleccionar un cierto epítipo j de entre r es $(1/r)$, y por lo tanto (μ/r) es la probabilidad de mutar hacia un epítipo j en particular y $(1-\mu/r)$ es la de no mutar al mismo.

Recordemos ahora que, un virus en el modelo es de longitud 16 (16 epítipos), por lo tanto, la probabilidad de que ninguno de sus epítipos mute a j es: $(1-\mu/r)^{16}$, y de que sólo uno mute a j es $(1-\mu/r)^{15}$, de modo que $(1-(1-\mu/r)^{16}) (V_n - V_n^j)$ es el número de los virus que no tenían el epítipo j y lo generaron en alguna de sus posiciones, y $\mu(1-1/r)(1-(1-\mu/r)^{15}) V_n^j$ es el número de los que lo mutaron y refabricaron en otro sitio.

Todas estas cantidades son proporcionales a la virulencia de la cepa, que denotaremos como γ , de aquí la ecuación para la población de células infectadas con todos los términos es:

$$I'_{n+1} = (1-\epsilon)(1-\delta T'_n) I'_n + \gamma \left[(1-\mu) + \mu \left(\frac{1}{r} + \left(1 - \frac{1}{r}\right) \left(1 - \left(1 - \frac{\mu}{r}\right)^{15}\right) \right) \right] I'_n + \gamma \left(1 - \left(1 - \frac{\mu}{r}\right)^{16}\right) (V_n - v'_n) \quad (11)$$

$$\text{Pero, } (1-\mu + \mu)^{16} = \sum_{k=0}^{16} \binom{16}{k} (1-\mu)^k \mu^k$$

y la probabilidad de crear j en k intentos: $1 - (1-1/r)^k$

De esta manera, la probabilidad de crear el epítipo j en k intentos a lo largo del cromosoma esta dada por

$$\sum_{k=0}^{16} \binom{16}{k} (1-\mu)^{16-k} \mu^k \left(1 - \left(1 - \frac{1}{r}\right)^k\right) = 1 - \left(1 - \frac{\mu}{r}\right)^{16} \quad (12)$$

Por otro lado, la cantidad de células T específicas contra la cepa j en el tiempo $n+1$ es: la del tiempo anterior, mas una cantidad proporcional a la población de virus de dicha cepa, menos las que “murieron” a causa de infección, cuya cantidad es proporcional al número total de virus y de células T en el tiempo n .

Despreciando fluctuaciones estocásticas y suponiendo una población infinita, tanto de virus como de células, ($r \gg \mu > 1$) el modelo matemático que representa al algoritmo genético es:

$$V_{n+1}^j = V_n^j + \alpha I_n^j - \rho V_n^j T_n^j \quad (13)$$

$$I_{n+1}^j = (1 - \varepsilon)(1 - \delta T_n^j) I_n^j + \gamma(1 - \mu) \left(1 - \frac{15\mu}{r}\right) V_n^j + \gamma(16\mu/r) V_n \quad (14)$$

$$T_{n+1}^j = T_n^j + \kappa I_n^j - \nu I_n^j T_n^j \quad (15)$$

en donde:

V_n^j es la población de virus con el epítipo j.

I_n^j es la población de células infectadas por virus que contiene el epítipo j

T_n^j es la población de células del sistema inmunológico específicas contra el epítipo j.

V_n es la población total de virus.

n es el tiempo.

$j = 1, \dots, r$ es la cepa del virus y

$r = 16N^2$ para autómatas $k=2$ y longitud N

En la ecuación (13), la cantidad de virus de la cepa j, en el tiempo $n+1$ (V_{n+1}^j) varía con respecto a la del tiempo anterior en $\alpha I_n^j - \rho V_n^j T_n^j$, en donde αI_n^j es la cantidad de virus generados por las células infectadas de la misma cepa. Cada célula produce α virus, y $\rho V_n^j T_n^j$ son los virus que “mueren” a causa de las células T específicas.

En (14) tenemos que la población de células infectadas disminuye por lisis, así como por el sistema inmunológico (δT_n^j); $1-\varepsilon$ es la proporción de células infectadas por la cepa j , que no han “muerto” debido a la producción de virus (por lisis), mientras que $1-\delta T_n^j$ es la porción que sobrevive a la acción del sistema inmunológico, en cantidad proporcional a la maquinaria inmunológica específica, que merma la cantidad de células infectadas, por lo que se incluye el producto de estos términos.

Las células se infectan en forma proporcional a la población de la cepa j y a la totalidad de virus, $(\gamma(1-\mu)(1-15\mu/r)V_n^j \gamma(16\mu/r)V_n)$.

Comparación del AG con el modelo de Nowak

Para comparar el comportamiento del Algoritmo Genético, con el modelo planteado por Nowak 1992, podemos forzar la muerte de las células infectadas en cada paso de tiempo, lo que significa que la lisis se produce por cada virus ($\varepsilon = 1$).

Las ecuaciones (13), (14) y (15) quedarán entonces como sigue:

$$T_{n+1}^j = T_n^j + \kappa V_n^j - \upsilon V_n T_n^j \quad (16)$$

$$V_{n+1}^j = V_n^j + \alpha\gamma[(1-\mu)(1-15\mu/r)V_{n-1}^j + (16\mu/r)V_{n-1}] - \rho V_n^j T_n^j \quad (17)$$

Eliminando el retraso por fabricación de virus a células infectadas, y de éstas a virus, el sistema que rige las poblaciones es el siguiente:

$$T_{n+1}^j = T_n^j + \kappa V_n^j - \upsilon V_n T_n^j \quad (18)$$

$$V_{n+1}^j = (1 - \rho T_n^j)V_n^j + \alpha\gamma[(1 - \mu - 15\mu/r + 15\mu^2/r)V_n^j + (16\mu/r)V_n] \quad (19)$$

Los cambios de población en un paso de tiempo se representan entonces en el sistema:

$$\Delta T^j = \kappa V^j - \upsilon V T^j \quad (20)$$

$$\Delta V = -(\rho T^j)V^j + \alpha\gamma[\alpha\gamma[(-\mu - 15\mu/r + 15\mu^2/r)V^j + (16\mu/r)V]] \quad (21)$$

Tenemos entonces, que las células T se incrementan en κV^j , y disminuye en $\upsilon V T^j$, debido a los encuentros con los virus.

De acuerdo con (21) el incremento en virus es $\alpha\gamma (16\mu/r) V$, y su decremento $\rho T^j V^j + \alpha\gamma [(-\mu \cdot 15\mu/r + 15\mu^2/r) V^j]$.

Comparando estos coeficientes con los de las ecuaciones (3), (4), tenemos que:

$$\begin{aligned} \kappa &\approx k, y \\ uV &\approx uV, \end{aligned} \tag{22}$$

Ademas,

$$\begin{aligned} \alpha\gamma (16\mu/r) V &\approx r, y \\ \rho T^j V^j + \alpha\gamma [(-\mu \cdot 15\mu/r + 15\mu^2/r) V^j] &\approx uvx_i \end{aligned} \tag{23}$$

Análisis de coeficientes de Liapunov

Una forma más clara de ver el comportamiento del sistema que constituye el modelo {(13), (14), (15)} es mediante sus coeficientes de Liapunov, ya que al analizar el sistema en un punto fijo, tendremos que si encontramos un coeficiente positivo entonces el sistema es estable, mientras que si el coeficiente es negativo corresponde a sistemas en los que los valores de las variables tienden a alejarse del correspondiente punto fijo.

Para ello se obtuvieron las ecuaciones para V_{n+1} , I_{n+1} y T_{n+1} , como representación de las cantidades totales de virus, células infectadas y células T, por lo cual se sumó sobre j el sistema {(13), (14), (15)} tomando las siguientes consideraciones:

$$\sum_j V_n^j = V_n \quad (24)$$

$$\sum_j V_n^j = 16V_n \quad (25)$$

$$\sum_j T_n^j = T_n \quad (26)$$

$$\sum_j I_n^j = I_n \approx 16T_n \quad (27)$$

$$\sum_j T_n^j I_n^j = 16T_n I_n, \text{ y } \sum_j V_n^j T_n^j = 16V_n T_n \quad (28)$$

El sistema que indica las poblaciones totales es el siguiente:

$$V_{n+1} = V_n + \alpha I_n + \rho V_n T_n \quad (29)$$

$$I_{n+1} = (1 - \epsilon)(1 - 16\delta T_n)I_n + \gamma(1 - \mu)\left(1 - \frac{15\mu}{r}\right)V_n + \gamma\left(\frac{16\mu}{r}\right)16V_n \quad (30)$$

$$T_{n+1} = T_n + 16kV_n - vV_n T_n \quad (31)$$

A partir del sistema {(29), (30), (31)}, se obtuvo:

$$\begin{pmatrix} \frac{\partial V_{n+1}}{\partial V_n} & \frac{\partial V_{n+1}}{\partial I_n} & \frac{\partial V_{n+1}}{\partial T_n} \\ \frac{\partial I_{n+1}}{\partial V_n} & \frac{\partial I_{n+1}}{\partial I_n} & \frac{\partial I_{n+1}}{\partial T_n} \\ \frac{\partial T_{n+1}}{\partial V_n} & \frac{\partial T_{n+1}}{\partial I_n} & \frac{\partial T_{n+1}}{\partial T_n} \end{pmatrix} = \begin{pmatrix} 1-16\rho T_n & \alpha & -\rho V_n \\ \gamma(1-\sigma)\left(1-\frac{15\mu}{r}\right) + (16\mu r) & (1-\varepsilon)(1-16\delta T_n) & 0 \\ k & 0 & 1 \end{pmatrix} \quad (32)$$

Linealizando alrededor de un punto fijo, como P_0 , en el que un paciente no presenta ni virus, ni células infectadas, sin importar la cantidad de células del sistema inmunológico, tenemos $(V_n, I_n, T_n) = P_0 = (0, 0, T_0)$, y si llamamos $\gamma' = \gamma(1-\mu)\left(1-\frac{15\mu}{r}\right) + \left(\frac{16\mu}{r}\right)$. En tal caso la matriz de coeficientes es:

$$\begin{pmatrix} 1-16\rho T_0 & \alpha & 0 \\ \gamma' & (1-\varepsilon)(1-16\delta T_0) & 0 \\ k & 0 & 1 \end{pmatrix} \quad (33)$$

y la ecuación característica correspondiente

$$(1-\lambda)\left[\lambda^2 + ((1-16\rho T_0) + (1-\varepsilon)(1-16\delta T_0))\lambda - \alpha\gamma'\right] = 0 \quad (34)$$

de donde

$$\lambda_1 = 1, \quad (35)$$

$$\lambda_2 = \frac{(1-16\rho T_0) + (1-\varepsilon)(1-16\delta T_0) + \sqrt{((1-16\rho T_0) + (1-\varepsilon)(1-16\delta T_0))^2 + 4\alpha\gamma'}}{2} \quad (36)$$

$$\lambda_3 = \frac{(1-16\rho T_0) + (1-\varepsilon)(1-16\delta T_0) - \sqrt{((1-16\rho T_0) + (1-\varepsilon)(1-16\delta T_0))^2 + 4\alpha\gamma'}}{2} \quad (37)$$

por lo cual $\lambda_3 > \lambda_1 > \lambda_2$ con $\lambda_1 = 1$.

Es importante mencionar que por tratarse de un sistema discreto, el neutro es la unidad, de aquí que estos coeficientes equivalen a un coeficiente positivo, uno negativo y uno cero para un sistema diferencial.

Lo que nos dice este resultado es que si

$$(1 - 16\rho T_0) + (1 - \varepsilon)(1 - 16\delta T_0) = 2\sqrt{\left(\left(1 - 16\rho T_0\right) + \left(1 - \varepsilon\right)\left(1 - 16\delta T_0\right)\right)^2 + 4\alpha\gamma'} \quad (38)$$

entonces, el sistema llega al punto fijo, es decir, el paciente se recupera al eliminar por completo virus y células infectadas. Pero de no cumplirse esto, el sistema se aleja del punto fijo, se incrementa así la cantidad de virus en el paciente y con ello las células infectadas; eso puede llevar al colapso del sistema inmunológico si la cantidad de virus alcanza la cantidad mínima letal (en el modelo equivale a alcanzar cantidad máxima de virus y/o de células infectadas), o bien se puede tener una latencia prolongada, vista como una dinámica en la que los virus y las células infectadas aumentan y disminuyen sin tocar los extremos (Recuperación y colapso).

Mediante el análisis de coeficientes de Liapunov, observamos que la dinámica del sistema, es similar a la que se presenta en los casos clínicos del VIH *in vivo* [M. A. Nowak, 1992].

4. Resultados

Se probaron dos casos especiales del modelo genético, para comprobar la validez de su representación vía las ecuaciones (13), (14) y (15), esos casos son:

- a). Sin muerte de células infectadas, ni acción por parte del sistema inmunológico y
- b). Sin la acción de células infectadas.

Para el primer caso, con el fin de obtener el comportamiento del sistema, al no reducir el número de virus ni el de las células que los producen, tenemos un sistema equivalente al que se tendría si fijáramos los parámetros ϵ y δ (que serían nulos), y eliminar lo relacionado al sistema inmunológico, de aquí que el sistema de ecuaciones modificado es el siguiente

$$I'_{n+1} = I'_n + \gamma(1-\mu)\left(1 - \frac{15\mu}{r}\right)I'_n + \gamma \frac{16\mu}{r} V'_n \quad (39)$$

$$V'_{n+1} = V'_n + \alpha I'_n \quad (40)$$

Básicamente se eliminan los términos que disminuyen las poblaciones con el fin de observar sólo su crecimiento. De estas dos ecuaciones, tenemos que

$$I'_n = \frac{V'_{n+1} - V'_n}{\alpha} \quad (41)$$

y por lo tanto, la población de células infectadas se puede representar por:

$$I'_{n+1} = \frac{1}{\alpha} V'_{n+2} - \frac{1}{\alpha} V'_{n+1} = \frac{1}{\alpha} V'_{n+1} - V'_n + \gamma(1-\mu)\left(1 - \frac{15\mu}{r}\right)I'_n + \gamma \frac{16\mu}{r} V'_n \quad (42)$$

mientras que la expresión que rige la población de virus es

$$V'_{n+2} - V'_{n+1} = V'_{n+1} + \left[\alpha\gamma(1-\mu)\left(1 - \frac{15\mu}{r}\right) - 1\right]V'_n + \alpha\gamma \frac{16\mu}{r} V'_n \quad (43)$$

De aquí que

$$V'_{n+2} = 2V'_{n+1} + \left[\alpha\gamma(1-\mu)\left(1 - \frac{15\mu}{r}\right) - 1\right]V'_n + \alpha\gamma \frac{16\mu}{r} V'_n \quad (44)$$

Ahora debe hacerse la suma sobre j para tener el sistema que determina la cantidad de todas las células infectadas y todos los virus sin importar la cepa. Para ello debe considerarse que un mismo virus puede contener diferentes determinantes antigénicos, es decir, un solo virus se puede estar contando hasta en 16 distintos j , ya que en el modelo un virus es de longitud 16, mientras que un determinante es de longitud 1.

Así pues

$$\sum_j V'_{n,j} = V'_{n,2} = 2I'_{n,1} + \left[16\alpha\gamma(1 - 1\mu) \left(1 - \frac{15\mu}{r} \right) - 1 \right] I'_n + \alpha\gamma \frac{16\mu}{r} V'_n \quad (45)$$

es decir,

$$V'_{n,2} - 2I'_{n,1} + V'_n = \alpha\gamma \left[1 - \mu + \frac{\mu}{r} - \frac{15(16)\mu^2}{r} \right] I'_n \quad (46)$$

Ecuación cuyo equivalente continuo es:

$$\frac{d^2V}{dt^2} = (\alpha\gamma) \left[1 - \mu + \frac{\mu}{r} - \frac{16(15)\mu^2}{r} \right] I' \quad (47)$$

por lo que

$$V(t) = Ae^{\sqrt{\alpha\gamma}t} + Be^{-\sqrt{\alpha\gamma}t} \quad (48)$$

Si comparamos este resultado con lo obtenido del Algoritmo Genético, con las modificaciones, podemos observar que, efectivamente, el modelo tiene un comportamiento como el expresado por la ecuación anterior, y las gráficas de la figura 3.8. Los datos experimentales corroboran pues, el buen funcionamiento del modelo.

En la tabla 4.1 se presentan unos resultados del AG, en donde t representa el tiempo, V la cantidad de virus, e I las células infectadas

t	V	I
0	71	1
10	106	8
35	500	35
44	900	70

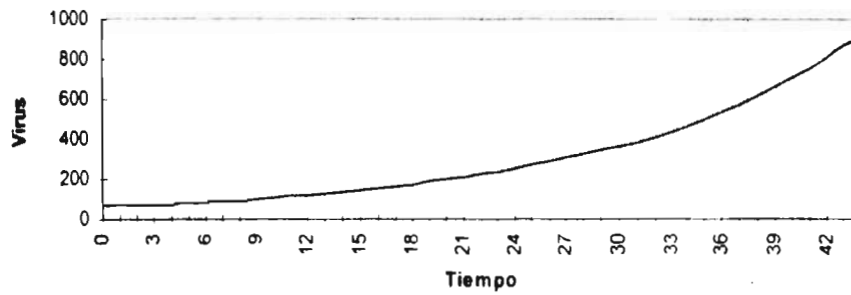
Tab. 4.1 Tamaños de poblaciones de virus V y células infectadas I a diferentes tiempos t , generados con el AG modificado.

Si evaluamos la función para los mismos valores de t , con $\sqrt{\alpha\gamma} = -0.06647$, $A=22.4007$ y $B=48.5992$, se obtiene la tabla 4.2.

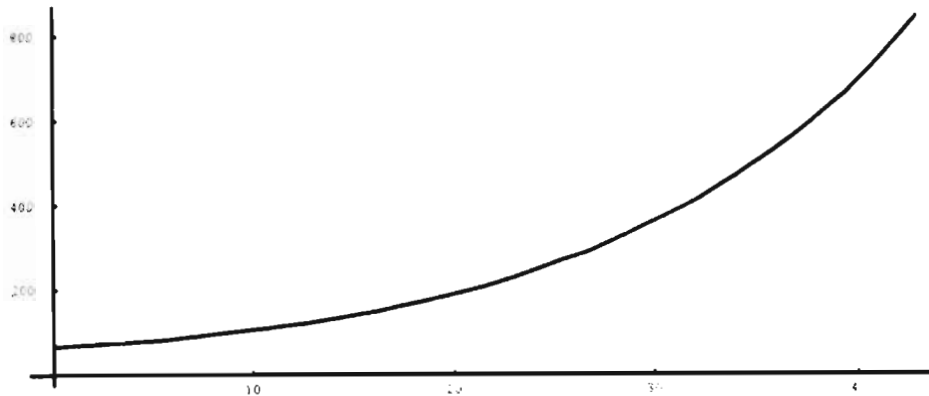
t	V
0	71
10	105.96
35	499.91

Tab. 4.2 Tamaños de población de virus V a diferentes tiempos t , obtenidos de la ecuación (41)

Comportamiento de Virus sin Cel. T



a). Gráfica de los datos obtenidos con el AG.



b). Gráfica del comportamiento de la población de virus sin la acción de las células T, regida por la ecuación.

$$V(t) = 22.4007e^{(-0.06647t)} + 48.5992e^{(0.06647t)}$$

Fig. 3.8. Comportamiento del sistema sin la acción del sistema inmunológico.

En el segundo caso, en el que la acción de las células infectadas se eliminó, lo que equivale a fijar el parámetro $\alpha = 0$ además de condicionar que la probabilidad de mutación a 0, esto para observar más claramente el comportamiento de la población de virus si no se pueden reproducir, de aquí obtenemos el siguiente sistema de ecuaciones:

$$T_{n+1} = T_n + kI_n - UT_n T_n \quad (48)$$

$$I_{n+1} = I_n - \rho I_n T_n \quad (49)$$

tomando la cota superior de correlación entre V_n y T_n , o lo que es lo mismo, si suponemos que existe una sola cepa dominante, se pueden despreciar los términos cuadráticos de las demás cepas, pues su cantidad es mínima, y como cada virus tiene 16 epitopos, tenemos que $\sum_i (I_n^i)^2 \approx 16I_n^2$, y entonces tenemos que:

$$T_n = \frac{T_n I_n}{16I_n} \quad \text{y} \quad (50)$$

$$\sum I_n = 16I_n \quad (51)$$

o bien

$$\sum I_n T_n = \frac{\rho T_n}{16I_n} \sum (I_n^i)^2 = \rho T_n I_n \quad (52)$$

de aquí podemos obtener el sistema de ecuaciones independientes de las cepas

$$T_{n+1} = T_n + 16kI_n - UT_n T_n \quad (53)$$

$$I_{n+1} = I_n - \rho T_n I_n \quad (54)$$

Para linealizar este sistema, obtendremos un punto fijo,

$$\frac{dT}{dt} = (16k)I - UT = 0 \quad (55)$$

$$\frac{dI}{dt} = -\rho T = 0 \quad (56)$$

de aquí que $V=0$ es un punto fijo.

Podemos plantear una ecuación para la población de virus:

$$\frac{1}{V} \frac{dV}{dt} = \rho T \quad (57)$$

$$\frac{d \log(V)}{dt} = -\rho T \quad (58)$$

$$V = Ae^{-\rho \int T dt} \quad (59)$$

$$\frac{d(VT)}{dt} = T \frac{dV}{dt} + V \frac{dT}{dt} \quad (60)$$

$$\frac{dT}{dt} = 16kAe^{-\rho \int T dt} - UATe^{-\rho \int T dt} \quad (61)$$

$$e^{\rho \int T dt} = \frac{16kA - UAT}{\frac{dT}{dt}} \quad (62)$$

$$\rho T = UA + (16kA - UAT) \frac{\frac{d^2 T}{dt^2}}{\left(\frac{dT}{dt}\right)^2} \quad (63)$$

$$\rho T \left(\frac{dT}{dt}\right)^2 = UA \left(\frac{dT}{dt}\right)^2 + (16kA - UAT) \frac{d^2 T}{dt^2} \quad (64)$$

si

$$T = Be^n \quad (65)$$

$$V = Ae^{-\rho \int T dt} \quad (66)$$

por lo que la población de virus se rige con la ecuación (67).

$$V(t) = Ae^{-\sqrt{16kA\rho} t} \quad (67)$$

Comparando los resultados que se obtuvieron del AG con las modificaciones y la ecuación (62), para $A=50$, $\sqrt{16kA\rho} = 0.014489$ Se obtuvo la tabla 4.3 .

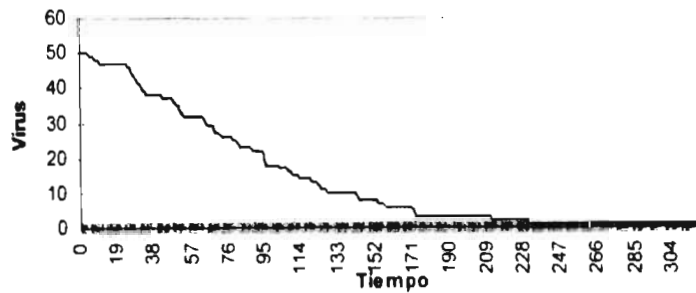
Resultados

Algoritmo Genético		ecuación (62)	
t	V	t	V
0	50	0	50
10	48	10	43.25
100	13	100	11.74
270	1	270	0.999

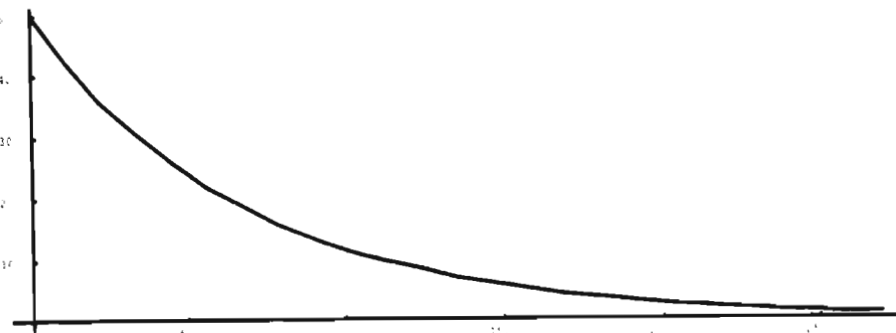
Tab. 4.3. Poblaciones de virus V a diferentes tiempos t, obtenidas por el AG y por la ecuación 65.

Resultados que también podemos observar en las gráficas de la figura 4.2.

Comportamiento de virus sin Cel. Inf.



a). Gráfica de los resultados obtenidos con el AG



b). gráfica de la ecuación $V(t)=50e^{-0.014489t}$

Fig. 4.2 Comportamiento del sistema sin la acción de células infectadas.

Regímenes Dinámicos

El sistema puede presentar tres regímenes distintos: 1) *Recuperación*, en el que la población de virus alcanza su mínimo (cero); 2) *Prolongación*, en el que ninguna de las poblaciones alcanza los máximos, o los mínimos en un tiempo determinado; y 3) *Colapso*, cuando los virus alcanzan su máximo, o bien las células T su mínimo.

Consideraremos a continuación cada uno de estos casos.

Recuperación

En las gráficas de las figuras 4.3 a 4.6 se observa un caso de recuperación del sistema. En la 4.3 se muestran simultáneamente las poblaciones de virus, células infectadas y células T, mientras que en el resto se aprecia cada una por separado.

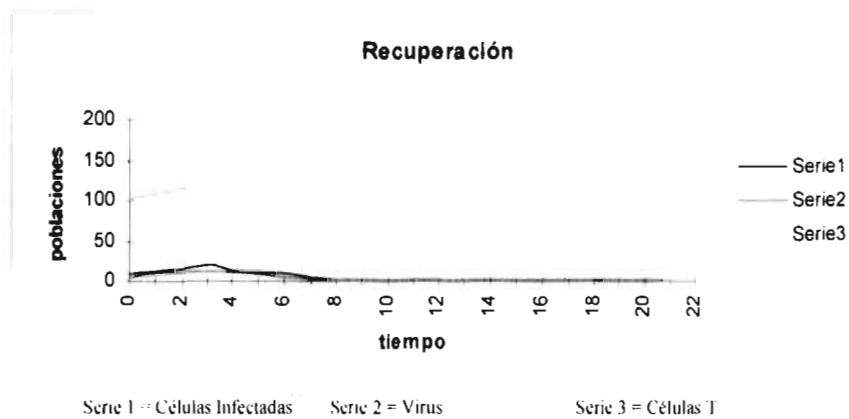


Fig. 4.3. Ejemplo del comportamiento del sistema en un caso de recuperación.



Fig. 4.4. Ejemplo del comportamiento de la población de células infectadas en un caso de recuperación.

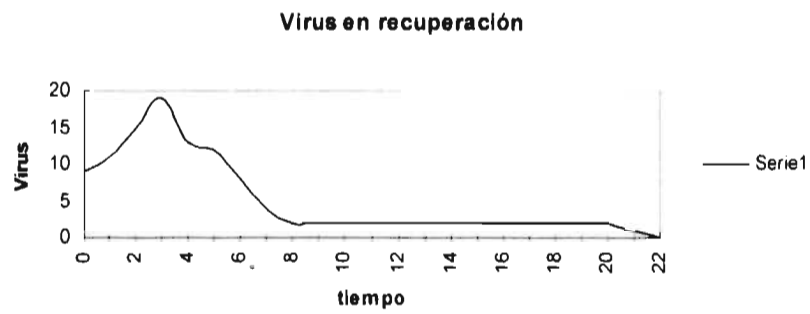


Fig. 4.5. Ejemplo del comportamiento de la población de virus en caso de recuperación.

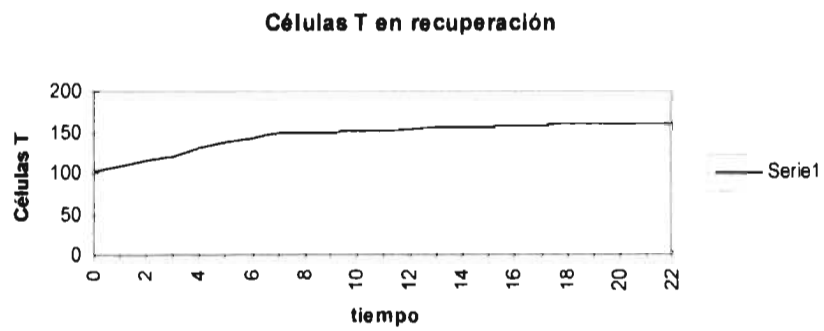


Fig. 4.6. Ejemplo del comportamiento de la población de células T en un caso de recuperación.

Latencia Prolongada

En un caso de latencia prolongada se puede observar que el sistema puede comportarse como en las siguientes gráficas. En la primera (4.7) se muestran las tres poblaciones juntas, y en las siguientes cada población por separado

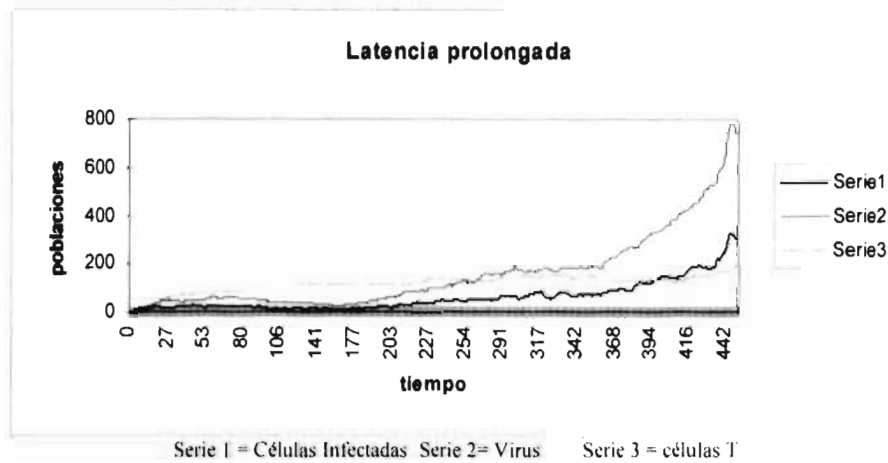


Fig. 4.7. Ejemplo del comportamiento del sistema en un caso de latencia prolongada.

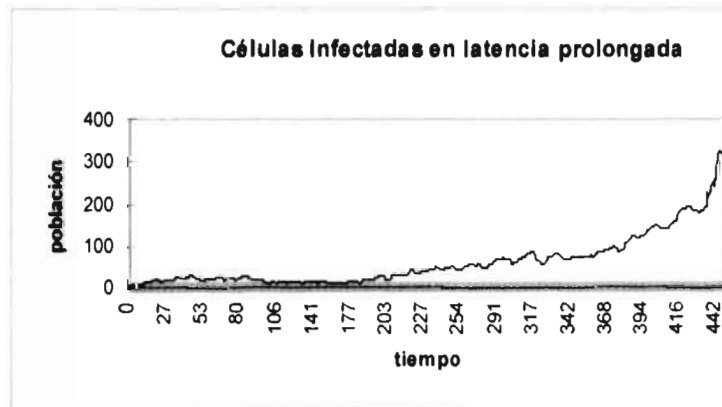


Fig. 4.8. Ejemplo del comportamiento del tamaño de la población de células infectadas en un caso de latencia prolongada.

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

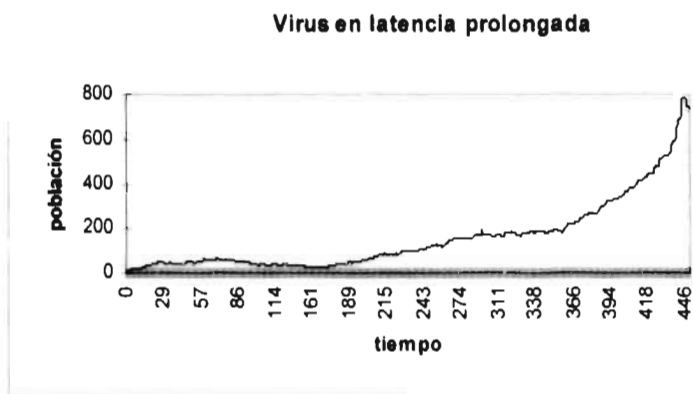


Fig. 4.9. Ejemplo del comportamiento de la población de virus en un caso de latencia prolongada.

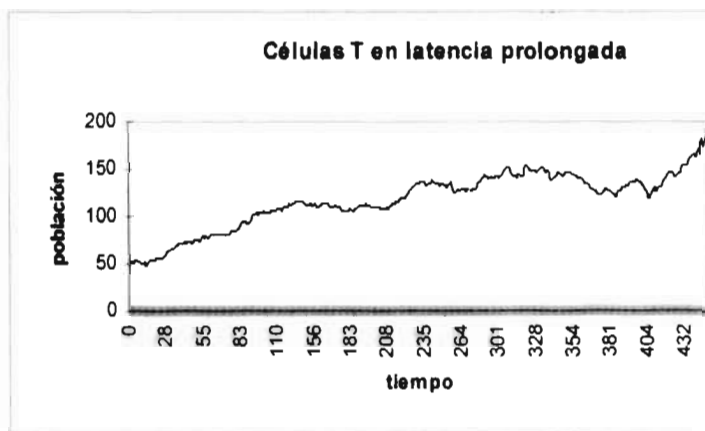


Fig. 4.10. Ejemplo del comportamiento de la población de células T en un caso de latencia prolongada.

Colapso del sistema

A continuación se muestra las gráficas de un caso en el que el sistema se colapsó: el paciente muere porque la población alcanza el máximo número de virus y/o células infectadas. En la primera gráfica se muestran simultáneamente las poblaciones de virus, células infectadas y células T, mientras que en las siguientes se puede observar cada una de las poblaciones a la vez.

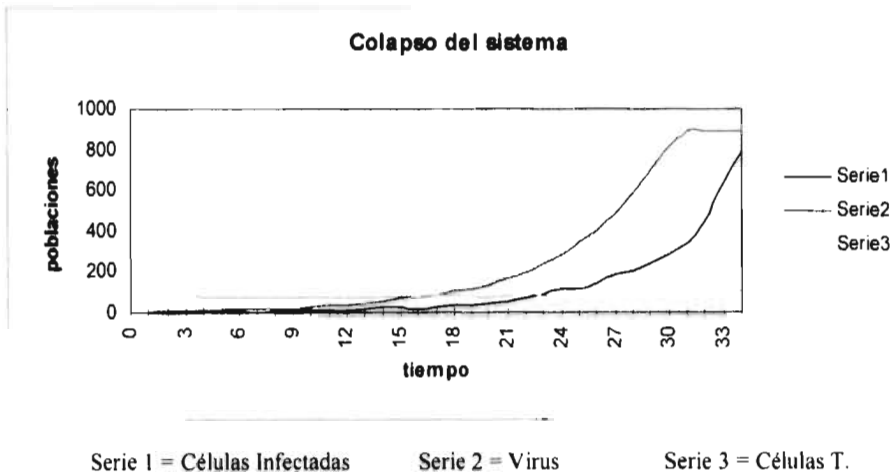


Fig. 4.11. Ejemplo del comportamiento del sistema en un caso de colapso.

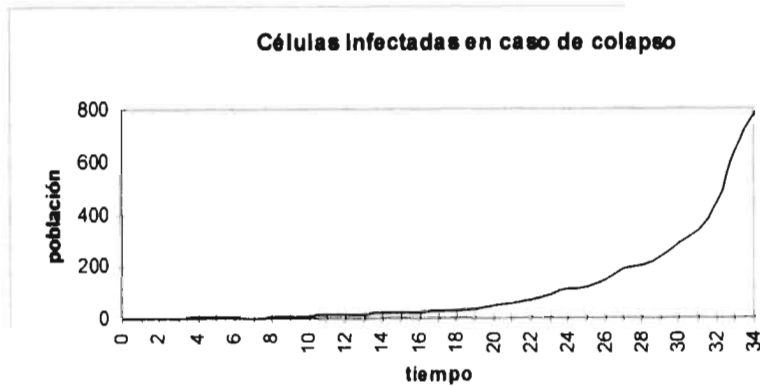


Fig. 4.12. Ejemplo del crecimiento de la población de células infectadas en un caso de colapso.

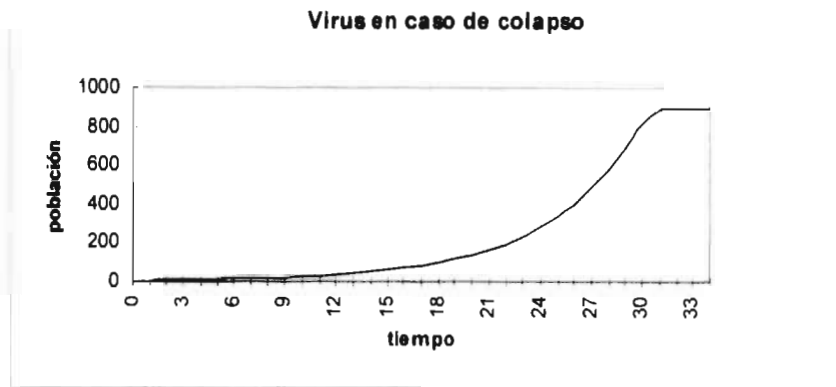


Fig. 4.13. Ejemplo del comportamiento de la población de virus en un caso de colapso.

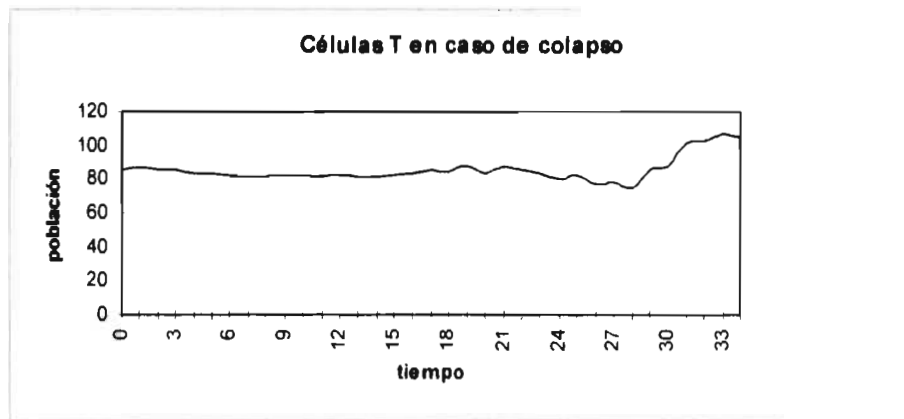


Fig. 4.14. Ejemplo del comportamiento de la población de células en un caso de colapso.

Comentarios

Una lista de las diferencias más importantes entre el modelo planteado por Nowak, 1992 y el AG, es la siguiente.

1. En el modelo matemático basado en ecuaciones diferenciales, que se utilizan para problemas continuos, se asumen poblaciones infinitas, mientras que en el Algoritmo Genético se plasma la característica de sistema discreto, así como un poblaciones finitas. En el presente trabajo se pretendió un modelo que se apegara más a la realidad, con resultados más próximos a lo que se observa en el problema de población del VIH in vivo.
2. En el AG se consideran los incrementos y decrementos en las poblaciones por mutaciones de una cepa a otra, y en el modelo matemático no. Es decir, mientras en el primero, un virus sólo participa en la producción de virus de la misma cepa, como se observa en la ecuación (1), en el algoritmo genético la población de una cierta cepa, se puede incrementar por mutaciones virus de una segunda cepa, y también los elementos de la cepa en particular pueden generar incrementos en otras distintas.

Aun cuando las mutaciones entre cepas, al momento de obtener la población total de virus, podrían omitirse, considero necesario reflejar el efecto que estas mutaciones provocan al permitir al virus cambiar su aptitud, al formar parte de otra cepa que puede ser mas no reconocida por el sistema inmunológico (más apta), o bien que puede estar asociada a una respuesta inmune completamente desarrolladas (menos apta).

3. En el modelo por AGs se introduce un retraso de una unidad de tiempo en la producción de virus, debido a la utilización de células infectadas, no considerado en el modelo matemático. Con este retraso se pretende plasmar el tiempo que le toma al sistema inmune el detectar un antígeno.
4. No se incluyeron depósitos de virus en el algoritmo genético, que mantienen una latencia prolongada planteada en las ecuaciones (5), (6), y (7); lo que significa una mejora de nuestro modelo .
5. Se toman en cuenta diferentes probabilidades para que los virus infecten células, y para ser detectados por el Sistema Inmunológico, ya que en la naturaleza no todos los virus tiene la misma patogenicidad, ni para todos los virus se genera una respuesta inmune con la misma intensidad.

Con base en lo anterior, podemos decir que:

Con el AG se espera encontrar un modelo más realista de la respuesta inmune, sobre todo en relación con los efectos de tamaño finito (punto 1) y la posibilidad de adaptación del pool genético a las debilidades de un individuo en particular (punto 5).

A pesar de las diferencias que hay entre los dos sistemas, se tiene un comportamiento similar, y en ambos casos, se pueden obtener los tres regímenes, que puede seguir una infección con VIH, ya que *grosso modo* ambos reflejan los aspectos más importantes del sistema formado por virus y sistema inmunológico. Aun que es importante mencionar que no es posible proporcionar estadísticas, que reflejen la similitud de estos modelos con el sistema real debido a falta de datos clínico.

Se pretende usar este trabajo en lo futuro, con la idea de que un punto fijo del autómata de Kauffman, equivale a una lista de epítopos expuestos o escondidos, y así tener un modelo de como la mutabilidad del virus permite escapar del sistema inmune también “escondiendo” un epítipo. Con éste modelo se esperaría analizar una evidencia preliminar de que la elección entre ARNs sinónimos juega un papel que favorece la habilidad del virus de generar mutantes capaces de escapar del sistema inmune [S. Vera, H. Waelbroeck, 1996].

Este trabajo pretende añadir características que al modelo matemático, que lo acerque al sistema que se pretende ejemplificar; además de que al ser un AG, se reduce el costo computacional por hacer una simulación del sistema.

Finalmente, quiero enfatizar que para el desarrollo del modelo, se hicieron grandes simplificaciones, como el hecho de incluir como únicos componentes del sistema inmunológico a las células T, o como el representar a un virus como un autómata celular de longitud 16; y que su aplicación esta enfocada a modelos poblacionales.

Apéndice A

Código del Algoritmo Genético

Variables y constantes del programa:

```
/* archivo varvir.h *
/*      s.v.n. */
/* definición de variables y constante globales para *
/* el modelo de VIH con A.G.      nov. 1996. */

/* constantes globales *
#define POPULATION 900          /* población de virus          */
#define NGENES 16              /* tamaño del virus          */
#define TIEMPODEVIDA 400      /* tiempo de vida maxmimo    */
#define CELIMAX 800           /* número maximo de celulas infectadas */
#define PCT 500               /* población de celulas T    */
#define LCT 1                 /* longitud de celulas T    */
#define BOOLEAN 16           /* número de reglas binarias */
#define PREC 0.0002          /* prob. de que un epitope sea identificado */
#define PRECK 0.03           /* prob. de que un epitope se identificado usado para la
                             reproducción de celulas T */
#define PMUT 0.001           /* prob. de mutación        */
#define ITERMAX 50           /* No. máximo de iteraciones para punto fijo */
#define VIDACEL 1            /* vida de una celula       */
#define NMAX 0.1             /* # max. de vir. que prod. una célula infectada en cada tiempo */
#define MCT 0.0003          /* cte. de prop. para la muerte de cel. T */
#define PATOGEN 0.1         /* patogenicidad            */

/* variables globales *

/* variables globales para rule16 */
/* ruleta de 16 */
#define P 50
#define LV 16
#define NUMREGLAS 16

/* semillas para el generador de números aleatorios *
int seedi, seedv, seedt, seed, seeda;

/* cromosomas */

/* virus */
int vr[POPULATION][NGENES]; /* vector de virus (reglas) */
int vw1[POPULATION][NGENES]; /* conexión 1 */
int vw2[POPULATION][NGENES]; /* conexión 2 */
float vv[POPULATION]; /* virulencia del virus p
/* poblacion nueva */
int vrn[POPULATION][NGENES]; /* vector de virus (reglas) */
int vw1n[POPULATION][NGENES]; /* conexión 1 */
int vw2n[POPULATION][NGENES]; /* conexión 2 */
float vvn[POPULATION]; /* virulencia del virus p
```

```

/* células infectadas */
int celir[CELIMAX][NGENES];          /* vector de celulas infectadas (reglas) */
int celiw1[CELIMAX][NGENES];        /* conexión 1 */
int celiw2[CELIMAX][NGENES];        /* conexión 2 */
int celie[CELIMAX];                  /* edad de la celula infectada */

/* poblacion nueva */
int celin[CELIMAX][NGENES];         /* vector de celulas infectadas (reglas) */
int celiw1n[CELIMAX][NGENES];       /* conexión 1 */
int celiw2n[CELIMAX][NGENES];       /* conexión 2 */
int celien[CELIMAX];                 /* edad de la celula infectada */

/* células T */
int celtr[PCT][LCT];                /* vector de celulas T (reglas) */
int celw1[PCT][LCT];                /* conexión 1 */
int celw2[PCT][LCT];                /* conexión 2 */
int celte[PCT];                      /* edad de la celula T */

/* poblacion nueva */
int celtrn[PCT][LCT];               /* vector de celulas T (reglas) */
int celw1n[PCT][LCT];               /* conexión 1 */
int celw2n[PCT][LCT];               /* conexión 2 */
int celten[PCT];                     /* edad de la celula T */

/* contadores */
int pob_vir;                          /* población de virus */
int pob_celI;                          /* población de celulas infectadas */
int pob_celT;                          /* población de celulas T */
int pob_virn;                          /* población de virus */
int pob_celIn;                         /* población de celulas infectadas (nueva) */
int pob_celTn;                         /* población de celulas T (nueva) */

/* "Good" random generator */

/* returns a uniform random number between 0.0 and 1.0
set idum to any negative value to initialize or reinitialize
the sequence */

#define M1 259200
#define IA1 7141
#define IC1 54773
#define RM1 (1.0 / M1)
#define M2 134456
#define IA2 8121
#define IC2 28411
#define RM2 (1.0 / M2)
#define M3 243000
#define IA3 4561
#define IC3 51349

```

```

.....
/*
                                CÓDIGO
*/
.....

```

Código de las funciones relacionadas con los números aleatorios

```

float rand1(int *idum)
{
    static long ix1, ix2, ix3;
    static float r[98];
    float temp;
    static int iff = 0;
    int j;
    if(*idum < 0 || iff == 0) {
        iff = 1;
        ix1 = (IC1 - (*idum)) % M1;
        ix1 = (IA1 * ix1 + IC1) % M1;
        ix2 = ix1 % M2;
        ix1 = (IA1 * ix1 + IC1) % M1;
        ix3 = ix1 % M3;
        for(j = 1; j <= 97; j++) {
            ix1 = (IA1 * ix1 + IC1) % M1;
            ix2 = (IA2 * ix2 + IC2) % M2;
            r[j] = (ix1 + ix2 * RM2) * RM1;
        }
        *idum = 1;
    }
    ix1 = (IA1 * ix1 + IC1) % M1;
    ix2 = (IA2 * ix2 + IC2) % M2;
    ix3 = (IA3 * ix3 + IC3) % M3;
    j = 1 + ((97 * ix3) / M3);
    if(j > 97 || j < 1) printf("rand1: error.n");
    temp = r[j];
    r[j] = (ix1 + ix2 * RM2) * RM1;
    if((temp == 0.0) || (temp == 1.0)) {
        if (temp == 0.0) temp = 0.0001;
        else if (temp == 1.0) temp = 0.9999;
    }
    return temp;
} /* end of rand1.c */

```

```

int aleatorio(int tope)
/* funcion que devuelve un entero aleatorio entre 1 y tope */
{
    int ia=0;
    float xrand;
    xrand = rand1(&seeda);
    ia = xrand*tope;
    return(ia);
}

```

Funciones de Reglas Booleanas

```
/* función que evalua una de las 16 reglas posibles •
/* parámetros: número de regla, primero y segundo operando •
int evarule16(int norule, int a, int b)
{int j, resul;
  j = a+2*b;
  resul = (norule&(1<-j))?!:0;
  return (resul);
}

/* función que evalua la existencia de un punto fijo en el genotipo •/
int eva_pto_fijo(int vraux[LV], int w1aux[LV], int w2aux[LV])
/* recibe un virus (reglas y 2 conecciones) •/
{
int fijo=0;      /* =0 no hay pto fijo, = 1 si hay pto. fijo •
int fen0[NGENES]; /* fenotipo 0 •/
int fen1[NGENES]; /* fenotipo uno •/

int ie,ke;
float je;

for (ie=0; ie<NGENES; ie++){ /* inicia aleatoriamente un fenotipo •
  je = rand1(&seed);
  if (je < 0.50) {fen0[ie] = 0;}
  else {fen0[ie] = 1;}
}
ke=0;
while ((fijo == 0) && (ke < ITERMAX)){
  fijo = 1; /* asumimos que hay punto fijo •
  ke++;
  for (ie=0; ie<NGENES; ie++){
    fen1[ie]=evarule16(vraux[ie], fen0[w1aux[ie]], fen0[w2aux[ie]]);
    if (fen1[ie] != fen0[ie]){
      fijo = 0;
      fen0[ie]= fen1[ie];
    }
  }
}
return(fijo);
}
```


Funciones relacionadas con las poblaciones

```
int ncte, nvc, ncic, ncie;
int nctm, nvm, ncim, ncimt;

int eva_pto_fjo(int vraus[NGFNES], int vw1aux[NGFNES], int vw2aux[NGFNES]);

/* Inicia todo en 0 */
void inicero(void)
{
    int ic,jc;

    for (ic=0; ic<POPULATION; ic++){ /* virus */
        for(jc=0; jc<NGENES; jc++){
            vm[ic][jc]=vr[ic][jc]=0;
            vw1n[ic][jc]=vw1[ic][jc]=0;
            vw2n[ic][jc]=vw2[ic][jc]=0;
        }
        vvn[ic]=vv[ic]=0.0;
    }
    pob_virn=pob_vir=0;
    for (ic=0; ic<CELIMAX; ic++){
        for(jc=0; jc<NGENES; jc++){
            celim[ic][jc]=celir[ic][jc]=0;
            celiw1n[ic][jc]=celiw1[ic][jc]=0;
            celiw2n[ic][jc]=celiw2[ic][jc]=0;
        }
        celien[ic]=celie[ic]=0;
    }
    pob_celln = pob_cell=0;
    for (ic=0; ic<PCT; ic++){
        for(jc=0; jc<LCT; jc++){
            celtrn[ic][jc]=celtr[ic][jc]=0;
            celtw1n[ic][jc]=celtw1[ic][jc]=0;
            celtw2n[ic][jc]=celtw2[ic][jc]=0;
        }
        celten[ic]=celte[ic]=0;
    }
    pob_celTn = pob_celT=0;
}
}
```

Funciones relacionadas con las poblaciones

```
int nctc, nvc, ncic, ncie;  
int nctm, nvm, ncim, ncimt;
```

```
int eva_pto_fjo(int vraux[NGFNFS], int vw1aux[NGFNFS], int vw2aux[NGFNFS]);
```

```
/* Inicia todo en 0 */
```

```
void inicero(void)
```

```
{  
  int ic,jc;  
  
  for (ic=0; ic<POPULATION; ic++){ /* virus */  
    for(jc=0; jc<NGENES; jc++){  
      vm[ic][jc]=vr[ic][jc]=0;  
      vw1n[ic][jc]=vw1[ic][jc]=0;  
      vw2n[ic][jc]=vw2[ic][jc]=0;  
    }  
    vvn[ic]=vv[ic]=0.0;  
  }  
  pob_virn=pob_vir=0;  
  for (ic=0; ic<CELIMAX; ic++){  
    for(jc=0; jc<NGENES; jc++){  
      celim[ic][jc]=celir[ic][jc]=0;  
      celiw1n[ic][jc]=celiw1[ic][jc]=0;  
      celiw2n[ic][jc]=celiw2[ic][jc]=0;  
    }  
    celien[ic]=celie[ic]=0;  
  }  
  pob_celln = pob_cell=0;  
  for (ic=0; ic<PCT; ic++){  
    for(jc=0; jc<LCT; jc++){  
      celtrn[ic][jc]=celtr[ic][jc]=0;  
      celtw1n[ic][jc]=celtw1[ic][jc]=0;  
      celtw2n[ic][jc]=celtw2[ic][jc]=0;  
    }  
    celten[ic]=celte[ic]=0;  
  }  
  pob_celTn = pob_celT=0;  
}
```

```

/* inicia aleatoriamente la población de células T */
void iniciaT()
{
int i, j;
int cint;          /* Cantidad inicial de células T */
cint = aleatorio(PCT/5);
for (i=0; i<cint; i++)
    for(j=0; j<LCT; j++){
        celtr[i][j]=aleatorio(BOOLEAN);
        celtw1[i][j]=aleatorio(NGENES);
        celtw2[i][j]=aleatorio(NGENES);
        celte[i]=0;      /* verificar edad de cel T */
    }
pob_celT = cint;
}

/* Crea una población inicial de virus */
void contagio()
{
int j;
int cinv;          /* Cantidad inicial de virus */
int vraux[NGENES], vw1aux[NGENES], vw2aux[NGENES];

cinv = aleatorio(POPULATION/9);
pob_vir=0;

while (pob_vir<cinv){
    for(j=0; j<NGENES; j++){
        vraux[j]=aleatorio(BOOLEAN);
        vw1aux[j]=aleatorio(NGENES);
        vw2aux[j]=aleatorio(NGENES);
    }
    if (eva_pto_fjo(vraux,vw1aux,vw2aux)!=0){ /* evalua punto fijo*/
/* si encuentro punto fijo = 1 , sino = 0 */
        for(j=0; j<NGENES; j++){
            vr[pob_vir][j]=vraux[j];
            vw1[pob_vir][j]=vw1aux[j];
            vw2[pob_vir][j]=vw2aux[j];
        }
        vv[pob_vir]=rand1(&seedt);
        vv[pob_vir] /= 100;
        pob_vir ++;
    }
}
}
}

```

```

/* producción de células infectadas */
void infeccion()
{
int ii, ir, ji;
int celiraux[NGENES];
int celiw1aux[NGENES];
int celiw2aux[NGENES];
float xrand;

    ii=0;
    while((ii < pob_vir) && (pob_cell < CELIMAX)) {
        xrand=rand1(&seedi);
        if (xrand <= vv[ii]){
            for(ji=0; ji<NGENES; ji++){
                celiraux[ji] = vr[ii][ji];
                celiw1aux[ji] = vw1[ii][ji];
                celiw2aux[ji] = vw2[ii][ji];
            }
            /* mutacion, Crea una nueva cepa de virus */
            nci++;
            xrand=rand1(&seedi);
            if (xrand <= PMUT){
                for(ir=0; ir<NGENES ;ir++){
                    celiraux[ir] = aleatorio(BOOLEAN);
                    celiw1aux[ir] = aleatorio(NGENES);
                    celiw2aux[ir] = aleatorio(NGENES);
                } /* aqui termina mutacion */
            }
            if (eva_pto_fjo(celiraux, celiw1aux, celiw2aux)!=0){
                for (jt=0; jt<NGENES; jt++){
                    celim[pob_celln][jt] = celiraux[jt];
                    celiw1n[pob_celln][jt] = celiw1aux[jt];
                    celiw2n[pob_celln][jt] = celiw2aux[jt];
                }
                celien[pob_celln]=0;
                pob_celln++;
            }
            ncie++;
        }
        ii++;
    }
}

```

```

/* incremento de la poblacion del virus por reproducción */
void repvir()
{
int i,j,n;
i=0;
while ((i<~pob_cell) && (pob_virn<~POPULATION)){
/* cada celula l produce NMAX virus */
n=0;
while ((celie[i] <~ VIDACEL) && (n<~NMAX) && (pob_virn<~POPULATION)){
for(j=0; j<~NGENES; j++){
vnr [pob_virn][j] = celir[i][j];
vw1n[pob_virn][j] = celiw1[i][j];
vw2n[pob_virn][j] = celiw2[i][j];
}
vvn[pob_virn]=randl(&seedv);
vvn[pob_virn] /=100;
if (vvn[pob_virn] > 0) nvc++;
celie[i] = (celien[i]++); n++; pob_virn++;
}
i++;
}
}

/* producción de celulas T */
void producCT()
{
float prob;
int ptov, novir;
int j;
novir=0;
while ((novir<pob_vir) && (pob_celTn<~PCT)) {
prob=randl (&seedt);
if (prob <= PRECK) { /* Probabilidad de ser reconocido */
ptov = aleatorio (NGENES-LCT+1);
for (j=0; j<~LCT; j++){
celtrn[pob_celTn][j] = vr[novir][ptov+j];
celtw1n[pob_celTn][j] = vw1[novir][ptov+j];
celtw2n[pob_celTn][j] = vw2[novir][ptov+j];
}
celten[pob_celTn] = 0;
pob_celTn++; ncte++;
}
novir++;
}
}

```

/* respuesta inmune, equivale a muerte de virus y a muerte de celulas infectadas */

```

void respImm()
{
    int ir, jr, kr, lr;
    int reconoce;
    float pr;
    for (ir=0; ir<pob_cell; ir++){ /* compara una cel T contra cada virus */
        for (kr=0; kr<pob_vir; kr++){ /* recorre cada virus */
            for (jr=0; jr<NGENES-LCT+1; jr++){ /* recorre cada gen del virus */
                reconoce = 1;
                for (lr=0; lr<LCT; lr++){ /* recorre cada gen de la cel T */
                    if ((celtr[ir][lr]!=vvn[kr][jr+lr]) ||
                        (celtw1[ir][lr]!=vw1n[kr][jr+lr]) ||
                        (celtw2[ir][lr]!=vw2n[kr][jr+lr])) reconoce = 0;
                }
                /* si el virus kr es reconocido, se destruye */
                if (reconoce == 1) {
                    pr=rand1(&seedv);
                    if (pr<PREC) {vvn[kr]=vvn[kr]-0.05;vvn[kr]=0;}
                    jr =NGENES;
                }
            }
        }
    }
    /* compara contra cada celula infectada */
    for (kr=0; kr<pob_cell; kr++){ /* recorre cada cel. I */
        reconoce = 1;
        for (jr=0; jr<NGENES-LCT+1; jr++){ /* recorre cada gen de la c. I */
            for (lr=0; lr<LCT; lr++){ /* recorre cada gen de la c. T */
                if ((celtr[jr][lr]!=celir[kr][jr+lr]) ||
                    (celtw1[jr][lr]!=celiw1[kr][jr+lr]) ||
                    (celtw2[jr][lr]!=celiw2[kr][jr+lr])) reconoce = 0;
            }
            /* si la cel. I kr es reconocida, se destruye */
            if (reconoce == 1) {
                pr=rand1(&seedv);
                if (pr<PREC) {celie[kr]=celien[kr]=VIDACEL;ncim--;}
                jr =NGENES;
            }
        }
    }
    celten[ir] = (celte[ir]++);
}
}

```

```

void muerteCT1()
{
    int ir,kr;
    float pmct; /* probabilidad de que muera una célula T */
    /* dCT = k*pob_vir - u*pob_vir*pob_celT */
    /* entonces mueren en forma proporcional a la cantidad de virus y células T
    pero de una cierta cepa */
    for (kr=0; kr<pob_vir; kr++){ /* recorre cada virus */
        pmct = rand1(&seed1);
        if (pmct <= MCT) {
            ir = aleatorio(pob_celT);
            celten[ir] = celte[ir]-TIEMPODEVIDA; /*se mata a la célula */
            nctm++;
        }
    }
}

void muerteCT()
{
    int ir,kr;
    float pmct; /* probabilidad de que muera una célula T */
    /* dCT = k*pob_vir - u*pob_vir*pob_celT */
    /* entonces mueren en forma proporcional a la cantidad de virus y células T
    pero de una cierta cepa */
    for (kr=0; kr<pob_vir; kr++){ /* recorre cada virus */
        for (ir=0; ir<pob_celT; ir++){
            pmct = rand1(&seedt);
            if (pmct <= MCT) {
                celten[ir] = celte[ir]-TIEMPODEVIDA; /*se mata a la cél. */
                nctm++;
            }
        }
    }
}

/* copia las nuevas poblaciones */
void cpnewpob(void)
{
    int ic,jc;
    pob_vir=pob_cel1=pob_celT=0;
    for (ic=0; ic<pob_vir; ic++){ /* actualización de virus */
        if (vvn[ic]>0) { /* si tiene virulencia */
            for(jc=0; jc<NGENES; jc++){
                vr[pob_vir][jc]=vm[ic][jc];
                vw1[pob_vir][jc]=vw1n[ic][jc];
                vw2[pob_vir][jc]=vw2n[ic][jc];
            }
            vv[pob_vir]=vvn[ic];
            pob_vir++;
        }
    }
}

/*actualización de células infectadas */
for (ic=0; ic<pob_celIn; ic++){

```

```

        if(celie[ic]!=VIDACEL1){
            for(jc=0; jc<NGENES; jc++){
                celir[pob_cell][jc]=celirn[ic][jc];
                celiw1[pob_cell][jc]=celiw1n[ic][jc];
                celiw2[pob_cell][jc]=celiw2n[ic][jc];
            }
            celie[pob_cell]=celien[ic];
            pob_cell++;
        } else ncimt++;
    }
}
/* actualización de células T */
for (ic=0; ic<pob_celTn; ic++){
    if(celte[ic]!=TIEMPODEVIDA){
        for(jc=0; jc<LCT; jc++){
            celtr[pob_celT][jc]=celtrn[ic][jc];
            celtw1[pob_celT][jc]=celtw1n[ic][jc];
            celtw2[pob_celT][jc]=celtw2n[ic][jc];
        }
        celte[pob_celT]=celten[ic]+1;
        pob_celT++;
    }
}
}

```

/* copia las poblaciones anteriores a las nuevas */

```

void cpantpob(void)
{
    int ic,jc;

    for (ic=0; ic<pob_vir; ic++){ /* actualización de virus */
        for(jc=0; jc<NGENES; jc++){
            vm[ic][jc]=vr[ic][jc];
            vw1n[ic][jc]=vw1[ic][jc];
            vw2n[ic][jc]=vw2[ic][jc];
        }
        vvn[ic]=vv[ic];
    }
    pob_virn=pob_vir;
}
/* actualización de células infectadas */
for (ic=0; ic<pob_cell; ic++){
    for(jc=0; jc<NGENES; jc++){
        celirn[ic][jc]=celir[ic][jc];
        celiw1n[ic][jc]=celiw1[ic][jc];
        celiw2n[ic][jc]=celiw2[ic][jc];
    }
    celien[ic]=celie[ic];
}
pob_celln = pob_cell;
/* actualización de células T */
for (ic=0; ic<pob_celT; ic++){
    for(jc=0; jc<LCT; jc++){
        celtrn[ic][jc]=celtr[ic][jc];
        celtw1n[ic][jc]=celtw1[ic][jc];
    }
}

```



```

        celw2n[ic][jc] = celw2[ic][jc];
    }
    celten[ic] = celte[ic];
}
pob_celTn = pob_celT;
}

void main(void)
{
    int tiempo=0;      /* n#mero de ciclos (edad) */
    int i,j;
    int veces;

    seed = -542;
    for(veces=0; veces<=10; veces++){
        tiempo=0;
        seedi = -1 * aleatorio(8363);
        seedt = -1 * aleatorio(71);
        seedv = -1 * aleatorio(92736);
        seed = -1 * aleatorio(4527);
        seeda = -1 * aleatorio(5296);
        inicero();

        iniciaT();
        contagio();
        printf("\tseedi\tseedt\tseedv\tseed\tseeda. \n");
        printf("\t%d\t%d\t%d\t%d\t%d\n",seedi,seedt,seedv,seed,seeda);
        printf("\ttiempo\tcelInf\tvirus\tcel. T\n");
        do{
            nctc = nvc=0;
            nctic = ncie=0;
            nctm = nvm=0;
            ncim = ncimt = 0;
            cpantpob();
            infeccion();
            repvir();          /* reproduccion del virus */
            producCT();
            resplnmul();

            /* y a muerte de de células infectadas */
            muerteCT();
            cpnewpob();
            printf("\t%d\t%d\t%d\t%d\n",tiempo,pob_celI,pob_vir,pob_celT);
            tiempo++;
        }while((tiempo<= TIEMPODEVIDA)&&(pob_celI < CELIMAX)&&(pob_vir > 0));
    }
}

```

Apéndice B

Tabla de las 16 reglas booleanas utilizadas en el trabajo

variables		Número de regla															
x	y	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Referencias

- [1] Artificial evolution and artificial intelligence, Evolution in Machine Learning Principles and techniques, Chapman Hall, 1989.
- [2] Coello C. A., Introducción a los Algoritmos Genéticos, Soluciones Avanzadas, Enero 1995,5-11.
- [3] Coffin J. M., HIV Population dynamics in vivo: Implications for genetic variation, pathogenesis, and therapy, Science vol. 267, January 1995, 483-489.
- [4] Cohen P., HIV team cracks receptor mystery, Science, May 1996, 16.
- [5] Gustav J. V. Nossal, Life, Death and the Immune System, Scientific American, September 1993, 21-30.
- [6] Holland J. H., Adaptation in natural and artificial systems, an introductory Analysis with Applications to Biology Control and Artificial Intelligence, MIT Press, Massachusetts, 1992.
- [7] Janeway A.Jr., How the Immune System Recognizes Invaders, Scientific American, September 1993, 41-48.
- [8] Kauffman S. A., Adaptation on Rugged Fitness Landscape, Lectures in the sciences of complexity, SFI studies in the sciences of complexity. Ed. D. Stein Addison-Wesley, Longman, 1989.
- [9] Kauffman S. A., Principles of Adaptation in Complex Systems, Lectures in the sciences of complexity, SFI studies in the sciences of complexity. Ed. D. Stein Addison-Wesley, Longman, 1989.
- [10] Kauffman S.A., Weinberg E.D., The NK model of rugged fitness landscape and its application to maturation of the immune-response, J. Theor. Biol. 141 (2), 1989, 211-245.
- [11] Koza J., Genetic Programming, on the Programming of computers by means of natural selection, Massachusetts Institute of Tecnology.
- [12] Levin S.A., Grefell B., Hastings A., Perlson A.S., Mathematical and Computational Challenges in Population Biology and Ecosystems Science, Science vol. 275, january 1997, 334-343.
- [13] Lockey R., Bukantz S., Fundamentos de Inmunología y Alergia, Interamericana, McGraw-Hill, Madrid, Esp. 1989.

- [14] Luca Peliti, Fitness Landscapes and Evolution, Dipartimento di Scienze Fisiche and Unità INFM Università "Federico II", Mostra d'Oltremare, Pad. 19 I-80125 Napoli, Italy, December 14, 1995.
- [15] Marrack P., Kappler J. W., How the Immune System Recognizes the Body, Scientific American, September 1993, 49-55.
- [16] Mejía Olvera M., Cantú Paz E., Algoritmos Genéticos: Descripción Básica y Fundamentos Matemáticos, Instituto Tecnológico Autónomo de México, Depto. Académico de Computación, Universidad Autónoma Metropolitana, En Línea, vol 1 No. 5
- [17] Miramontes Vidal P., Autómatas Celulares, Algoritmos Genéticos y Evolución Molecular, Departamento de Matemáticas, Facultad de Ciencias, UNAM, México D.F.
- [18] Nicotini C., Modeling and Analysis in Biomedicine, World Scientific, Singapore, 1984.
- [19] Nowak M.A., McMichael A.J., How HIV Defeats the Immune System, Scientific American, August, 1995, 42.
- [20] Nowak M.A., Mini Review Variability of HIV Infections, J. Theor. Biol. 155 (1992) 1-20.
- [21] Preston B. D., Poiesz V, Fidelity of HIV-1 Reverse Transcriptase, Science, Vol 242, 1168-1171
- [22] Roberts J. D., The Accuracy of Reverse Transcriptase from HIV-1, Science, Vol 242, 1171-1173
- [23] Roitt I. M., Inmunología Esencial, 3ª. Edición Madrid, JIMS, S.A.
- [24] Schechter M. T., Craib K. J. P., Le T. N., Montaner J. SG., Douglas B., Sestak P., Willoughby B., O'Shaughnessy M. V., Susceptibility to AIDS progression appears early in HIV infection, AIDS. 144 (3), 1990, 185.
- [25] Schwartz R. H., T Cell anergy, Scientific American, August 1993, 48-54.
- [26] Solé R. V., Luque B., Phase transitions and antichaos in generalized Kauffman networks, Physics Letters A, January 1995, 331-334.
- [27] Stephens C. R., Waelbroeck H., Analysis of the effective degrees of freedom in genetic algorithms, Preprint ICN-UNAM-96-08 (adap-org/9611005).
- [28] Stites D.P., Stobo J.D., Inmunología Básica y Clínica, J.V. Wells

[29] Tjian R., Molecular Machines that control genes, Scientific American, February 1995, 38-45.

[30] Vera S., Waelbroeck H., Symmetry Breaking and Adaptation: the genetic code of retroviral "env" proteins, Preprint ICN-UNAM-96-09 (adap-org/9610001).

[31] Villalpando Hoyo A. M., Síndrome de Inmundeficiencia Adquirida (SIDA), tesis de licenciatura, Facultad de Odontología UNAM, 1988.

[32] von Boehmer H., Kisielow P., How the Immune system learns about self, Scientific American, October 1991, 50-59.