

03063



UNIVERSIDAD NACIONAL AUTONOMA DE
MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

13

PROGRAMACIÓN GENÉTICA MULTI-RAMAS EN EL MODELADO
Y PREDICCIÓN DE DATOS CLIMATOLÓGICOS.

T E S I S

QUE PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS

P R E S E N T A:

CARLOS OLIVER MORALES

DIRECTOR: KATYA RODRÍGUEZ VÁZQUEZ

México, D.F.

TESIS CON
FALLA DE ORIGEN

2002



Universidad Nacional
Autónoma de México



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.

A mis seres queridos

Agradecimientos

A mi familia por todo el cariño y apoyo que me han dado sin el cual no podría haber realizado este trabajo.

A la Dra. Katya Rodríguez Vázquez por haber dirigido este trabajo.

A la UNAM, profesores y compañeros.

Así mismo, agradezco el apoyo financiero del Consejo Nacional de Ciencia y Tecnología (CONACyT) a través del proyecto J34900-A, y al Dr. Abraham Rojano Aguilar quien proporcionó los datos analizados.

Índice general

1	Introducción.	1
1.1	Introducción	1
1.2	Objetivos y hipótesis	2
1.3	Contribuciones.	3
1.4	Estructura de la tesis	4
2	Técnicas convencionales	5
2.1	Introducción	5
2.2	Técnicas	5
2.3	Ajuste de curva	6
2.4	Regresión	7
2.5	Definición de modelo	8
2.6	Identificación de modelos	10
2.6.1	Diseño del experimento	11
2.6.2	Datos	12
2.6.3	Selección del modelo	12
2.6.4	Criterios de evaluación	14
2.6.5	Validez del modelo	15
2.6.6	problemas con modelación tradicional	15
3	Técnicas no tradicionales	17
3.1	Lógica borrosa	17
3.1.1	Introducción	17
3.1.2	Definiciones de lógica borrosa	18
3.1.3	Operaciones con conjuntos borrosos	18
3.1.4	Aplicaciones de la lógica borrosa	19
3.1.4.1	Modelado borroso	20
3.1.4.1.1	Modelo de Takagi – Sugeno	21
3.2	Redes Neuronales	24
3.2.1	Introducción	24
3.2.2	Estructura y características de las redes neuronales	24
3.2.3	Aprendizaje de las redes neuronales	27
3.2.4	Método de retro-propagación	28
3.2.5	Aplicaciones de las redes neuronales	28
3.2.6	Redes neuronales evolucionadas	29
3.2.7	Redes tipo ART	30
3.3	Computación evolutiva	33
3.3.1	Algoritmos genéticos	34
3.3.1.1	Pasos del algoritmo genético	36
3.3.1.2	Parámetros de control	37
3.3.1.3	Aplicaciones de los algoritmos genético	37
3.3.1.4	Teorema de esquema	38
3.3.2	Programación genética	39
3.3.2.1	Pasos de la programación genética	40
3.3.2.2	Parámetros de control	42
3.3.2.3	Aplicación de la programación genética	43
3.4	Comparativo entre técnica	43
4	Descripción de técnicas	45
4.1	Método de tradicional	45
4.1.1	Introducción	45
4.1.2	Método por Descomposición	45
4.2	Programación genética	47
4.2.1	Representación propuesta	47
4.2.2	Operadores genéticos	50

4.2.3	Evaluación de los individuos	51
4.3	Teoría de la resonancia adaptativa	52
4.3.1	Algoritmos genéticos: Determinación de parámetros	52
4.3.2	Evaluación de los individuos	53
4.3.3	Operación de la red ART2	54
4.4	Sistema de inferencia difuso con algoritmos genéticos	56
4.4.1	Algoritmo genético: Determinación del antecedente	56
4.4.2	Operación del sistema difuso de inferencia	59
4.4.3	Cálculo de coeficientes	60
4.5	Casos de prueba	60
4.5.1	Tradicional	60
4.5.2	Programación genética	61
4.5.3	Red ART2	61
4.5.4	Algoritmos genéticos y sistemas de inferencia difuso	62
5	Resultados	63
Caso 0:	Tradicional	63
Caso 1:	Programación Genética	64
Caso 2:	Programación Genética	65
Caso 3:	Programación Genética	67
Caso 4:	Red ART2	70
Caso 5:	Red ART2	71
Caso 6:	Red ART2	72
Caso 7:	GAFIS	74
6	Discusión, Trabajos Futuros y Conclusiones	78
6.1	Discusión	78
6.2	Conclusiones	84
6.3	Trabajos Futuros	85
	Bibliografía	86
	Anexos	90
A):	Modelos obtenidos con programación genética, con función objetivo de la norma infinito	90
B):	Errores de las mejores propuestas de cada corrida, Caso 1	90
C):	Diferentes normas obtenidas por grupo, en predicción a largo plazo, Caso 1	91
D):	Modelos obtenidos con programación genética, con función objetivo Norma 1	91
E):	Errores de las mejores propuestas de cada corrida, Caso 2	92
F):	Diferentes normas obtenidas por grupo en predicción a largo plazo, Caso 2	92
G):	Modelos obtenidos con programación genética, con función objetivo de la norma infinito y norma 1	93
H):	Errores de las mejores propuestas de cada corrida, Caso 3	94
I):	Diferentes normas obtenidas por grupo, en predicción a largo plazo Caso 3	94
J):	Relación de los parámetros de vigilancia y pesos obtenidos con algoritmos genéticos	
Caso 4:		95
K):	Resultados de las normas durante el proceso de prueba y durante el proceso de entrenamiento. No de patrones y tiempos para el Caso 4	95
L):	Relación de los parámetros de vigilancia y pesos obtenidos con algoritmos genéticos	
Caso 5:		96
M):	Resultados de las normas durante el proceso de prueba y durante el proceso de entrenamiento. No de patrones y tiempos para el Caso 5	97
N):	Relación de los parámetros de vigilancia y pesos obtenidos con algoritmos genéticos	
Caso 6:		97
O):	Resultados de las normas durante el proceso de prueba y durante el proceso de entrenamiento. No de patrones y tiempos para el Caso 6	98
P):	Normas por corrida durante cada fase, Caso 7	98

Capítulo 1

1.1 Introducción

La identificación de sistemas es un problema de búsqueda del modelo que describe mejor al fenómeno en estudio. Se puede recurrir a herramientas de las matemáticas y de las ciencias para obtener un modelo. Existen diversas formas para describir modelos dependiendo del fenómeno o problema; una forma de representarlos es por medio de modelos matemáticos los cuales utilizan un sistema de ecuaciones lineales, no lineales o diferenciales. Las computadoras son utilizadas comúnmente aprovechando únicamente su capacidad para realizar cálculos precisos y rápidamente. Otra alternativa es considerar a las computadoras basándose en su capacidad para resolver problemas. Durante el proceso de identificación existen etapas que requieren de acciones mecánicas y puramente de cálculo, pero también existen etapas en las cuales los humanos toman decisiones sobre la base del estado del arte de la ciencia, sus experiencias personales y sus apreciaciones objetivas y subjetivas. Etapas como la determinación del modelo dependen más de las decisiones humanas que de un proceso puramente algorítmico. Las actividades que son difíciles de expresar en forma algorítmica tradicionalmente no utilizan las computadoras. Sin embargo, una idea antigua es también resolver problemas que requieran más que únicamente cálculo. La inteligencia artificial ha generado diversas técnicas cuyo interés en las computadoras no solo se basa en su capacidad de cálculo, si no también aprovechan su capacidad para resolver problemas. Dentro de estas técnicas están las redes neuronales artificiales, sistema borroso de inferencia y computación evolutiva. Durante los últimos años se han ido generando diferentes implementaciones para problemas con diferentes dominios proporcionando evidencias desde un punto de vista pragmático, de su eficacia tanto en forma individual como en versiones híbridas para una aplicación específica. La finalidad de este trabajo es mostrar la implementación de una representación de polinomios por medio de ramas para PG (Programación Genética) y comparar con otras técnicas no convencionales la precisión y complejidad de las soluciones propuestas. Las otras técnicas no convencionales sobre las que se realizará la comparación son las RN (Redes Neuronales) siendo la red ART (Teoría de Resonancia Adaptativa) la tomada como ejemplo y FIS (Sistema Borroso de Inferencia), en específico una versión de GAFIS (Sistema Borroso de Inferencia con Algoritmos Genéticos).

La implementación de estas técnicas se aplican a un problema de modelado de series de datos climatológicos. El clima ha sido uno de los fenómenos que el hombre ha tratado de predecir desde la antigüedad hasta nuestros días. La principal razón de este gran interés es la importancia del clima

en todas las actividades humanas. El clima afecta directamente a los seres vivos y el hombre no es la excepción. Además de la actividad biológica, también tiene efectos en sus actividades económicas y sociales. Las propuestas de predicción van desde las puramente subjetivas hasta las formales y científicas, teniendo diferente precisión y alcance. Otra razón para seleccionar estos fenómenos como caso de estudio es el hecho de ser fenómenos que bajo las técnicas tradicionales se consideran fenómenos impredecibles. En el trabajo de Lorenz de la década de los 60's (Lorenz, 1963), el cual trataba de proponer una reducción del número de ecuaciones en el sistema de ecuaciones diferenciales que predecía las condiciones del tiempo, detectó inestabilidad en el sistema la cual no se debía a errores en cuanto a la determinación del modelo sino a las propiedades del fenómeno, conocida como caos. Los fenómenos caóticos son impredecibles, por lo menos a largo plazo (Koza, 1992).

Basado en lo anteriormente descrito, los problemas reales que se desean resolver son dos que están relacionados con estaciones meteorológicas. (1) El primero surge cuando un equipo de la estación falla, por lo que las medidas que se deberían realizar se pierden irremediablemente, no es posible regresar el tiempo para intentar tomar los datos nuevamente. Por lo tanto, se busca obtener estimaciones basándose en sus datos históricos tanto de la variable que se desea estimar como alguna otra variable climática que pueda influir. (2) El segundo problema es buscar la predicción de la variable del estado del tiempo antes de que el fenómeno se mida. Esta información deberá proporcionar una medida del funcionamiento del equipo, que oportunamente permita detectar fallas. La variable que se consideró como ejemplo es la temperatura ambiente. Esta tesis, está dirigida, principalmente, al segundo problema y con las soluciones obtenidas resolver el primero

1.2 Hipótesis y Objetivos

Implementación de una propuesta alternativa de representación de PG, la cual nos permita resolver problemas de modelado e identificación de sistemas proporcionando modelos eficientes y al mismo tiempo simples. Asimismo, realizar un estudio y análisis comparativo con otras técnicas no tradicionales, tales como, redes neuronales, y sistemas borrosos de inferencia, presentando la utilidad de las mismas

Objetivos:

- a) Realizar una versión propia de la implementación de Programación Genética con la representación de polinomios por multi-ramas.

- b) Comparación con redes neuronales artificiales y sistemas borrosos de inferencia para la predicción de series de tiempo, con el fin de detectar las ventajas y desventajas de cada una de ellas
- c) Aplicar las técnicas no tradicionales para la predicción de variables climatológicas para poder monitorear el buen desempeño de la estación y detectar fallas oportunamente

1.3 Contribuciones

- 1) Representación de polinomios por ramas, propuesta alternativa para la programación genética
- 2) Construcción de una caja de herramientas para la predicción de series de tiempo en MATLAB la cual considera :
 - a) Técnicas tradicionales
 - i). Método de descomposición
 - b) Técnicas no tradicionales
 - i). Programación genética, representación polinomial con multiramas.
 - ii) GAFIS.
 - iii) ART2
- 3) Presentación de artículos en eventos de nivel nacional e internacional:
 - i) Nacional:

Oliver-Morales, C y K. Rodríguez-Vázquez (2001) Técnicas No Tradicionales en la Predicción de Sistemas Dinámicos, *3er. Encuentro de Computación. Aguascalientes, México ENC 01* Vol. 1, pp 73-82.

- ii) Internacional:

Oliver-Morales, C y K. Rodríguez-Vázquez (2002) Estructura de Árbol vs Estructura Polinomial con Programación Genética en el Modelado de Variables Climatológicas, *1er Congreso Español de Algoritmos Genéticos y Bioinspirados*, a celebrarse del 6 al 8 de Febrero del 2002, Mérida, España (artículo aceptado)

1.4 Estructura de la tesis

En el capítulo 2 se presentan las ideas y conceptos básicos sobre el modelado de sistemas, así como una breve descripción de las técnicas convencionales para resolver dichos problemas. Antecedentes y descripción general de las técnicas no tradicionales estudiadas en este trabajo (*PG programación genética*, *LB lógica borrosa* y *RN redes neuronales*) se detallan en el capítulo 3. En el capítulo 4 se muestra la implementación de las técnicas no convencionales de PG por medio de ramas, la red ART2 con capacidad predictiva y GAFIS. Finalmente, los resultados son presentados en el capítulo 5 y la discusión, trabajos futuros y conclusiones generales en el capítulo 6.

Capítulo 2

2 Técnicas convencionales

En este capítulo se presenta una revisión de los métodos que tradicionalmente se utilizan para resolver problemas de recuperación y predicción. Para el problema de recuperación (completar) de datos, convencionalmente se utilizan técnicas de interpolación, colocación o aproximación. Para el problema de predicción además de la aproximación, se realiza por medio de la identificación del sistema, modelado del sistema y del método de la descomposición. Estas técnicas son un pequeño ejemplo de la gran variedad de técnicas que se han desarrollado para resolver problemas como los planteados en este trabajo.

2.1 Introducción

La colocación puede hacerse por medio de polinomios, funciones splines, funciones racionales o series de Fourier entre otras posibles. La mayoría de los métodos numéricos se basan en el ajuste polinomial, dada sus propiedades de fácil derivación e integración. Los errores de medición son una causa para no interpolar datos medidos, para lo cual se recomiendan métodos de ajuste de curvas.

La regla del *trapecio* es la base de varios modelos numéricos de colocación lineal. La interpolación lineal en dos dimensiones requiere de dos puntos conocidos $g(x)$ es la línea a estimar presentada en la ecuación [2.1] dado dos puntos conocidos $(a, f(a))$ y $(b, f(b))$, donde el primer elemento pertenece a la variable independiente y el segundo elemento pertenece a la variable dependiente (Nakamura, 1992).

$$g(x) = \frac{(b-x)f(a)}{b-a} + \frac{(x-a)f(b)}{b-a} \quad [2.1]$$

2.2 Técnicas convencionales

La medición de datos solo puede realizarse en forma discreta aunque el fenómeno sea continuo. Se requiere de técnicas que estimen los puntos intermedios de los valores discretos. Los procesos de aproximación no tienen la finalidad de que la curva propuesta pase por los puntos de soporte sino

que el error existente entre los valores aproximados y los observados de acuerdo con una cierta métrica sea mínimo. La métrica de aproximación más común es la de mínimos cuadrados. La otra visión es adaptar una curva exactamente sobre los puntos medidos (puntos de soporte); al conjunto de estas técnicas se les conoce como colocación. La colocación más simple y subjetiva puede ser la graficación considerándola como un método pre-computacional. Dos ejemplos donde los métodos de colocación son utilizados son, el análisis de la tendencia y la prueba de hipótesis. Para el primer caso, éste es un proceso que determina el patrón de los datos para realizar predicciones. Cuando la tendencia se utiliza para la predicción o pronósticos de los datos se le llama extrapolación. Para la prueba de hipótesis se propone un modelo matemático que se compara con los datos medidos, buscando los coeficientes que mejor describan el comportamiento de dichos datos (Chapra y Canale 1990).

La interpolación se aplica para estimar los valores intermedios entre los puntos de datos de soporte, siendo cero el error de la estimación en los puntos de soporte. Para una función de orden polinomial n , como la ecuación [2.2], donde $f(x)$ es la función a estimar, a_i los coeficientes del polinomio y x es la variable independiente, si existen $n+1$ puntos existe uno y solo un polinomio de orden n que pasa por n puntos. Sin embargo, hay una gran variedad de formas en las cuales el polinomio se puede expresar tales como: Lagrange, Chevyshev y Newton, entre otras (Nakamura, 1992).

$$f(x) = a_0 + a_1x^1 + a_2x^2 + \dots + a_nx^n \quad [2.2]$$

El método de Newton pertenece a los métodos iterativos que van aproximando los valores estimados a los valores reales al considerar polinomios de mayor grado. El método para determinar la representación explícita de una interpolación polinomial desde una tabla se le conoce como el método de diferencias divididas (Burden y Faires, 1988).

Cuando los puntos existen a igual distancia y los errores se distribuyen homogéneamente, los polinomios de Chevyshev son más adecuados que los polinomios de Lagrange o el método de Newton. Los polinomios de Chevyshev son ampliamente utilizados en subrutinas matemáticas y cálculos numéricos generales (Nakamura, 1992).

2.3 Ajuste de curva

Los datos que se obtienen a partir de las mediciones del fenómeno de interés tienen fluctuaciones. Esto se debe a errores aleatorios del sistema de medición aplicado al comportamiento comúnmente estocástico del sistema en observación. Entre más datos se tengan mayor es la precisión de la curva ajustada. Un método que minimiza la desviación entre los valores estimados y los valores reales es

el método de mínimos cuadrados. Un ajuste de curvas también puede realizarse por medio de una combinación lineal de funciones conocidas. La curva ajustada a los puntos puede escribirse como la ecuación [2.3], donde $g(x)$ es la función a estimar, a_i son coeficientes, $f_i(x)$ funciones conocidas y n el número de funciones (Nakamura, 1992)

$$g(x) = a_1 f_1(x) + a_2 f_2(x) + \dots + a_n f_n(x) \\ = \sum_{n=1}^n a_n f_n(x) \quad [2.3]$$

2.4 Regresión

El problema a resolver es determinar cuál es el comportamiento que realmente ocurre entre dos puntos el cual generalmente es desconocido. Las medidas de comparación entre las estimaciones ($f(x_i)$) con los modelos y los valores reales (y_i) son: el error residual (e_i) (ecuación [2.4]), la suma del error residual mostrado en la ecuación [2.5]. Una mejor medida es la suma del error absoluto (ecuación [2.6]) y la suma del error cuadrático [2.7].

$$e_i = y_i - f(x_i) \quad [2.4] \qquad \sum_{i=1}^n |e_i|, \quad |e_i| = |y_i - f(x_i)| \quad [2.6]$$

$$\sum_{i=1}^n e_i \quad [2.5] \qquad \sum_{i=1}^n (e_i)^2 = \sum_{i=1}^n (y_i - f(x_i))^2 \quad [2.7]$$

Para el proceso de mínimos cuadrados del modelo expresado en la ecuación [2.8], donde z_0, z_1, \dots, z_m son funciones diferentes, a_i los coeficientes y e es el error, empieza expresando dicho modelo en forma matricial. La expresión [2.9] es la expresión matricial donde $\{E\}$ es el vector de los errores, $\{Y\}$ el vector de los valores esperados, $\{Z\}$ la matriz que representa los valores de z y $\{A\}$ es el vector de coeficientes. El objetivo es encontrar el vector $\{A\}$ que minimice el error cuadrático S . n es el número de puntos de soporte y m el número de coeficientes a determinar. Las expresiones [2.11 - 2.14], ejemplifican el proceso para determinar el vector $\{A\}$.

Modelo:

$$y = a_0 z_0 + a_1 z_1 + \dots + a_m z_m + e \quad [2.8]$$

$$[Z] = \begin{pmatrix} z_{01} & z_{11} & \dots & z_{m1} \\ z_{02} & z_{12} & \dots & z_{m2} \\ \vdots & \vdots & & \vdots \\ z_{0n} & z_{1n} & \dots & z_{mn} \end{pmatrix} \quad \{A\} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix} \quad \{Y\} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad [E] = \{Y\} - [Z]\{A\} \quad [2.9]$$

$$S = \sum_{i=1}^n (y_i - \sum_{j=0}^m a_j z_{ij})^2 \quad [2.10]$$

Nota : i - ésima observación
 j - ésimo coeficiente

$$\{Y\} - [Z]\{A\} = 0 \quad [2.11]$$

$$[Z]\{A\} = \{Y\} \quad [2.12]$$

$$[Z]^T [Z]\{A\} = [Z]^T \{Y\} \quad [2.13]$$

$$\{A\} = ([Z]^T [Z])^{-1} [Z]^T \{Y\} \quad [2.14]$$

La obtención de $[Z]^T [Z]^{-1}$ puede realizarse por medio de una descomposición LU, descomposición de Cholesky y por la inversión de la matriz. En forma analítica la minimización de las sumatorias se realiza por el método de mínimos cuadrados, donde se desea minimizar la suma de las diferencias al cuadrado entre los valores estimados y los valores reales.

2.5 Definición de modelos

La palabra *modelo* es usada en muchas situaciones para describir sistemas, consecuentemente hay fuertes diferencias de opinión sobre cual es el apropiado uso de la palabra modelo. Todo modelo puede tener propósitos específicos tales como los pronósticos y el control. El modelo solo debe considerar en detalle lo que absolutamente sea necesario para satisfacer esos propósitos. La premisa inicial en la construcción de modelos es: no por que el fenómeno sea complejo el modelo necesariamente tiene que ser complejo. Los modelos son tradicionalmente deterministas o estocásticos. La principal desventaja de los modelos deterministas es que no existe la posibilidad de generar diferentes modelos con el mismo conjunto de datos empíricos. El criterio de mínimos cuadrados puede ser usado para la comparación de modelos. Modelos deterministas que satisfacen el criterio de mínimos cuadrados convenientemente, no siempre obtienen resultados aceptables para predicción y control. La utilidad de un modelo esta íntimamente relacionada con su capacidad de generalización. La capacidad de generalización es cuando el modelo generado con n observaciones del sistema no varía mucho con otro modelo construido con las siguientes n observaciones. La construcción de modelos en un contexto de probabilidad permite comparar diferentes modelos lo

Este tipo de ruido no siempre es considerado en los modelos pero también es importante incluirlo (Kashyap y Ranachandra, 1976). Existen diferentes formas para expresar la interrelación de las diferentes variables de los sistemas y donde el grado de formalismo puede variar. Existen modelos que describen las relaciones utilizando gráficos y tablas. Por otro lado, existen los modelos matemáticos que describen las variables en términos de expresiones matemáticas, tales como las ecuaciones diferenciales también conocidas como modelos analíticos. Hay que diferenciar claramente entre los modelos matemáticos y la realidad. El modelo no puede representar *exactamente la realidad*. Sin embargo, para fines prácticos, los modelos son considerados como sistemas reales. Los modelos pueden ser construidos sobre la base de los datos observados o bien desarrollados por la experiencia. Una forma de entender más claramente fenómenos complejos es dividiendo el fenómeno en partes y tratar de crear para cada parte un modelo. La intuición y la experiencia también proporcionan ideas para crear modelos estructurados por medio de diagramas de bloques (Ljung, 1999) y la utilización de estructuras gráficas para representar modelos. Las reglas gráficas pueden representar fácilmente los conocimientos humanos, presentando ligas causales o por el contrario ligas irrelevantes entre las variables. La estructura de la gráfica puede mostrar la transición de estados del modelo. También pueden mostrar la transición con los diferentes sub-modelos donde cada nodo de la gráfica es un modelo (Johansen y Murray, 1997)

2.6 Identificación de modelos de sistemas

La identificación de modelos puede tener varios pasos como lo muestra la Figura 2.2. Cuando se desea identificar un modelo es esencial la definición de los objetivos de la modelación y el propósito del modelo; por ello, es importante identificar los siguientes puntos: diseño del experimento y la adquisición de los datos, el preproceso de los datos, el análisis de conocimientos a priori, las máquinas de modelación (estructura e identificación de parámetros), la simplificación con la reducción del modelo, la validación, el análisis e interpretación del modelo. La pre-estructuración de modelos basados en regímenes de operación es de especial interés en algoritmos con aprendizaje. La descomposición en regímenes de operación es una forma de disminuir la complejidad de los sistemas, buscando identificar aspectos que permitan determinar la estructura del modelo local. El análisis y el uso de conocimientos a priori es esencial en la búsqueda de una buena estructura del modelo (por ejemplo, el conocimiento es acerca de los objetivos del problema, los efectos del ambiente y los requerimientos de robustez). Si no se tiene información a priori, el aprendizaje y la adaptación es un buen sustituto (Johansen y Murray, 1997)

cual da un significado preciso de las palabras significativo e insignificante (Kashyap y Ranachandra, 1976).

El inferir modelos desde observaciones y estudio de propiedades es el quehacer de la ciencia, caracterizado por hipótesis, leyes de la naturaleza y paradigmas. El sistema es un objeto en el cual las variables de diferentes tipos sufren una interacción y producen señales observables. En los sistemas podemos diferenciar 4 tipos de señales: las señales de entrada (u), las señales de salida (y), el ruido que se puede medir (w) y el ruido que no se puede medir (v). La Figura 2.1 muestra un diagrama con estas señales (Ljung, 1999). Hay variables en los sistemas que no permiten que sean medidos durante los procesos, pero también son parte importante en la determinación de la dinámica del sistema ya que intervienen en forma crítica sobre la definición de los estados. Por otro lado no todos los estados de un sistema pueden llegar a conocerse (Yang, Chen y Wang, 2000)

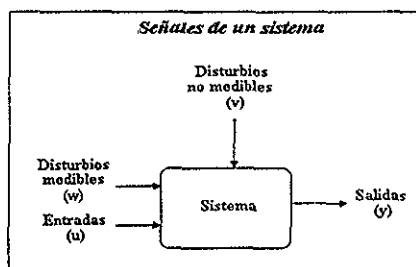


Figura 2.1: Señales del sistema (Ljung, 1999).

Tradicionalmente existe una división arbitraria de las variables en dos grupos, las entradas y las salidas. Esta división está basada en la división de variables independientes (entradas) y las variables dependientes (salidas). Sin embargo, existen sistemas donde existe una retroalimentación donde las salidas son también entradas, por lo que al construir modelos se debe contar con todas las variables donde las relaciones causales serán inferidas por el modelo.

Una importante consideración en la modelación es el rol del error en las observaciones. Estos errores se deben a baja precisión de la medida de los instrumentos y esquemas. Esto es especialmente cierto en instrumental que se encuentra al aire libre como el utilizado en estaciones meteorológicas. Si $y(t)$ es la medida de una salida del sistema, el valor real de la salida se representa como $y(t)$ y $n(t)$ el ruido teniendo la siguiente relación,

$$y(t) = y(t) + n(t) \quad [2.15]$$

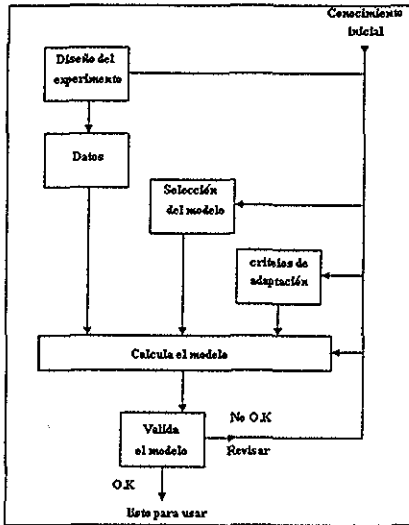


Figura 2.2: Pasos en el proceso de modelado e identificación del sistema (Ljung, 1999).

2.6.1 Diseño del experimento

Durante el diseño del experimento se identifican las entrada y salidas que describirán al sistema a modelar, así como cual será su uso y que resultados debe tener. Las variables de los sistemas describen los estados del sistema, por lo que se determina que variables deben de ser medidas, como se pueden evaluar y que limitaciones físicas se pueden tener. La fuente de información que puede ser utilizada son los datos experimentales, los modelos lineales incompletos, los conocimientos cualitativos y las heurísticas (Molina, Isasi, Berlanga y Sanchis, 2000; Bahat, Inbar y Schenieder, 2000; Emami, Goldenberg y Türkensen, 2000). La información recolectada puede ser usada para determinar parámetros, como en los modelos de caja blanca, o la estructura como en los modelos de caja negra (Babuška, Verbrugen y Can, 1999). Otra consideración en esta etapa es la descomposición de un modelo complejo, la cual se basa en la estrategia de divide y vencerás. La descomposición puede realizarse en 5 dimensiones: por componentes físicos, basada en el fenómeno, en términos de series de expansión, por objetivos y en regimenes de operación. Cuando los procesos requieren de modelos de elevada complejidad y alta no linealidad, los métodos analíticos tradicionales son de poca utilidad (Hans, 1999).

2.6.2 Datos

La adquisición de datos es el primer paso del proceso para generar modelos. Esto se realiza haciendo una colección de patrones de señales de los sensores los cuales corresponden a características particulares del proceso. Las diferentes señales corresponden a un estado del sistema. Para realizar el análisis se requiere de una conversión entre el valor del sensor y el estado del sistema. Esta conversión lo realizan las llamadas herramientas de extracción. También se debe considerar la adquisición de los datos en el diseño del experimento; los datos experimentales son usados para identificar estructuras y parámetros de un modelo. Estos datos muestran la frecuencia y distribución de situaciones particulares, que tan significativas son, los niveles de ruido y la definición de regiones. La precisión que tengan los datos está relacionada con lo que represente el modelo y el uso del modelo. Los datos obtenidos deben cubrir todo el rango de operación y debe existir información suficiente sobretodo donde el sistema sea más complejo.

2.6.3 Selección del modelo

Los modelos matemáticos pueden describirse como continuos o discretos, aglomerados o distribuidos, deterministas o estocásticos y lineales o no lineales. La Figura 2.3 describe un modelo matemático general, donde y es la salida, n_a los tiempos de retraso en la salida, a_i los coeficiente de salida, $i = 1, \dots, n_a$, u es la entrada, n_b los tiempos de retraso en entrada, b_j los coeficientes de entrada y $j = 1, \dots, n_b$, e_k es el error donde c_k son los coeficientes asociados con el error, $k = 1, \dots, n_c$, θ vector renglón de coeficientes y $\phi(t)$ el regresor

$$y(t) + a_1 y(t-1) + \dots + a_{n_a} y(t-n_a) = b_1 u(t-1) + \dots + b_{n_b} u(t-n_b) + \dots + c_1 e(t-1) + \dots + c_{n_c} e(t-n_c)$$

Para simplificar:

$$\theta = [a_1, \dots, a_{n_a}, b_1, \dots, b_{n_b}, c_1, \dots, c_{n_c}]^T$$

$$\Phi(t) = [-y(t-1), \dots, -y(t-n_a), u(t-1), \dots, u(t-n_b), e(t-1), \dots, e(t-n_c)]^T$$

$$y(t) = \Phi(t)^T \theta$$

Figura 2.3: Modelo general (Ljung, 1999).

En la Tabla 2.1 se describen algunos modelos con sus polinomios característicos. *AR* se refiere a autoregresivo, *MA* se refiere a medias móviles y *X* a las variables exógenas (Ljung, 1999)

Tabla 2.1: Ejemplos de modelos (Ljung, 1999).

Polinomio usado	Nombre de la estructura (modelo)	Descripción
<i>B</i>	<i>FIR</i>	<i>finite impulse response</i>
<i>AB</i>	<i>ARX</i>	<i>Autoregresivo con variables exógenas</i>
<i>ABC</i>	<i>ARMAX</i>	<i>Autoregresivo, medias móviles con variables exógenas</i>
<i>AC</i>	<i>ARMA</i>	<i>Autoregresivo, medias móviles</i>

La información más general define el orden del sistema y la forma del modelo que debe ser identificada (*ARX*, *ARMAX*, *ARMA*, etc.) Ciertas áreas del espacio de estado pueden ser modeladas usando modelos físicos bien estructurados y pueden ser incluidos directamente como modelos locales. La razón principal es que no existe la información suficiente que cubra todo el espacio de estados. Algunos modelos pueden completar la información, pero la principal desventaja puede ser obtener el modelo dado su costo y su precisión (Johansen y Murray, 1997).

Una forma de describir las series de tiempo es por medio de sus parámetros y la determinación de sus tendencias. Las tendencias se pueden dividir en tres grupos, la tendencia creciente, la tendencia decreciente y la estática. La tendencia es el primer aspecto que se desea determinar cuando se busca el modelo (Yang, Chen y Wang, 2000). En una serie histórica de datos, conocida como *serie de tiempo*, existen cuatro patrones básicos: la horizontalidad, la tendencia, la estación y el ciclo. La horizontalidad se refiere a los casos estacionarios donde no existe tendencia. El patrón de estación se presenta cuando una serie de tiempo fluctúa de acuerdo a un factor dependiente del periodo del año. El patrón de ciclo son fluctuaciones que ocurren más lentamente. El patrón de tendencia existe cuando una serie de tiempo tiende a disminuir o aumentar los valores medios con el tiempo. La tendencia es un movimiento en un periodo extenso de tiempo, hay diferentes estilos de crecimiento. Los estilos de crecimiento más comunes son el de una recta, el geométrico, la curva *pearl-reed*, la parabólica, la logarítmica y *gompertz* (Yamane, 1979). Por otro lado, las series de tiempo se clasifican por su comportamiento como: i) pseudo-periódicas, ii) cuasi-periódicas, iii) periódicas y iv) aperiódicas. Las series de tiempo pseudo-periódica son series que describen fenómenos con un comportamiento fuertemente cíclico que se aproxima a un periodo de tiempo pero no tienen indicación de crecimiento. Ejemplo de estas series es el número de manchas solares a través del año, el número de lobos canadienses vistos a través del año y el flujo de agua del río en los diferentes meses. Un periodograma puede mostrar la forma de los picos de la frecuencia (ecuación [2.16])

$$f = 1/T \quad [2.16]$$

donde f es la frecuencia y T el número de ciclos por unidad de tiempo. Otro grupo de series de tiempo son las aproximadamente periódicas con crecimiento. Ejemplo de esta serie de tiempo son las ventas mensuales. El tercer grupo de series de tiempo son aquellas que exhiben un crecimiento o decaimiento pero no presentan componentes periódicos. Ejemplos de estas series son el precio de las acciones de IBM y el producto interno del país. También existen las series de tiempo sin periodicidad o crecimiento. Este tipo de series tienden a aumentar durante cierto periodo de tiempo y tienden a disminuir durante otros. Estas oscilaciones son completamente irregulares. Ejemplos de estas series son la forma de onda del habla y las temperaturas observadas a cada instante. (Kashyap y Ranachandra, 1976).

2.6.4 Criterios de evaluación

Las formas de evaluar el error generalmente se calculan sobre la base del sesgo o en términos de varianza; sin embargo, estas son muy sensibles al ruido. Otra forma de evaluarlo más consistente se muestra en la ecuación [2.17] (Nelles, 2001)

$$\text{Generalización error} \cong \text{Variancia ruido} \left(1 + \frac{\text{Número de Parámetros}}{\text{Número de observaciones}} \right) \quad [2.17]$$

La evaluación de la precisión del modelo se realiza por medio del AIC (*criterio de información de Akaike*) mediante la ecuación [2.18], donde N es el número de observaciones, $\hat{y}(t)$ indica el valor estimado de $y(t)$ que es el valor esperado y n es el máximo grado del modelo (Morimoto, Susuki y Hashimoto, 1997)

$$V = -\frac{1}{2} \sum_{t=1}^N \varepsilon^2(t), \quad \varepsilon^2(t) = (y(t) - \hat{y}(t))^2$$

$$AIC = \ln\{(1 + 2n / N)V\} \quad [2.18]$$

2.6.5 Validez del modelo

Una vez obtenido el modelo el siguiente paso consiste en evaluar el desempeño bajo estados del sistema que no fueron utilizados durante el proceso de creación. Como medida del desempeño se pueden utilizar los mismos criterios de evaluación, en datos nuevos, a los que se utilizaron durante la creación del modelo.

2.6.6 Problemas con la modelación tradicional

Modelar y predecir el comportamiento de sistemas es un proceso complejo debido a que generalmente son caracterizados por un gran número de variables, parámetros, interacciones y una limitada colección de información (Zitang, Ohm y Mühlenbein, 1997). Los modelos de sistemas basados en herramientas matemáticas convencionales no son muy adecuados para sistemas mal definidos o con incertidumbre. Por el contrario, un sistema de inferencia borrosa puede modelar aspectos de conocimiento y razonamiento humano sin emplear un análisis cuantitativo preciso (Jang, 1993). La predicción puede ser obtenida por medio de métodos analíticos pero tienen varias limitaciones, generalmente es difícil tener una buena precisión para fenómenos complejos, como son reacciones, la flexibilidad y la fricción. También el número de parámetros internos puede ser grande, los cuales no siempre son fáciles de calcular con la precisión necesaria o requiere de grandes cantidades de información. Por último, los modelos construidos son difíciles de usar en tiempo real debido a su complejidad (Emami, Goldenberg y Türkensen, 2000). Los métodos tradicionales de pronósticos utilizan series de tiempo para realizar sus predicciones. Sin embargo, la no linealidad y complejidad de los sistemas han generado nuevos métodos de predicción (Srinivasan, Liew y Chang, 1994). La complejidad y la incertidumbre es el mayor cuello de botella para aplicar métodos tradicionales (Babuška, Verbrugen y Can, 1999). Además de las desventajas antes mencionadas, todas las técnicas convencionales ven a las computadoras como una herramienta donde se implementa la solución del problema; la otra alternativa es considerar la capacidad de las computadoras para resolver el problema.

En el siguiente capítulo trataremos el tema de técnicas no tradicionales que también obtienen resultados iguales o mejores para resolver el problema, evitando en parte las desventajas de los métodos convencionales. Un aspecto importante de estas técnicas no convencionales es su tendencia a ser autónomas, autonomía en específico de las decisiones de los usuarios finales. Sin embargo, esta autonomía no se ha concretado completamente; por ejemplo, sigue dependiendo del preproceso

de la información y de decisiones del usuario final. Es por ello que resulta importante dedicar tiempo a la investigación de algoritmos más inteligentes.

Capítulo 3

3 Técnicas no convencionales

En este capítulo se presentan las diferentes técnicas no convencionales cuyas implementaciones se realizaron en este trabajo. Cabe aclarar que existen otras técnicas o versiones de las mismas, pero en este capítulo solo nos limitaremos a revisar las técnicas de Lógica Borrosa, Redes Neuronales y Programación Genética.

3.1 Lógica borrosa

3.1.1 Introducción a la lógica borrosa

Convencionalmente suponemos que los parámetros de un modelo describen en forma precisa la realidad, inequívocamente y sin ambigüedades. Para los modelos debe existir una fuerte relación entre el lenguaje formal y el lenguaje del dominio que permitan la identificación sin confusión. Un conjunto clásico está definido como una colección de elementos o de objetos $x \in X$ el cual puede ser finito contable o incontable, o bien infinito. Cada elemento puede pertenecer o no al conjunto. Los conjuntos se describen por enumeración, por la descripción analítica mencionando las instancias y condiciones de pertenencia o usando la función característica (Kaufmann y Gupta, 1985). La palabra fuzzy (borroso) fue introducida por Zadeh, en su trabajo Fuzzy Sets (Zadeh, 1965), aplicándose a conjuntos donde existe una función de pertenencia que determina, para cada elemento del conjunto, un valor de pertenencia $\mu \in [0, 1]$, para describir conceptos no precisos, ambiguos, que están relacionados con lo indefinido, dudoso, donde no se requieren precisiones altas y/o donde la opinión de los expertos es importante (Bahat, Inbar y Schenieder, 2000; Khoo, Ang y Zhang, 2000; Zimmermann, 1991; Kaufmann y Gupta, 1985; Morimoto, Susuki y Hashimoto, 1997). Existen diferentes modelos de un sistema, donde se aplica un armazón que es utilizado para la combinación de la información cuantitativa y de los datos numéricos con cualidades y conocimientos heurísticos. El objetivo es hacer sistemas más amigables y la traducción del conocimiento a la máquina en forma más manejable, lo cual se consigue con conjuntos borrosos, que son utilizados para describir variables en términos lingüísticos (Dash, Liew, Rahman y Dash, 1985; Kaufmann y Gupta, 1985; Emami, Goldenberg y Türkensen, 2000).

Los sistemas borrosos pueden identificarse como sistemas de clasificación y sistemas de aproximación. El problema de clasificación consiste en hacer un mapeo entre las entradas a los patrones discretos. El consecuente determina a que patrón pertenece un vector de entrada, mientras que en los problemas de aproximación el mapeo entre el vector de entrada se hace a un valor real (Babuška, Verbrugen y Can, 1999; Nelles, 2001; Dash, Liew, Rahman y Dash, 1985)

3.1.2 Definiciones de lógica borrosa

Una forma de expresar los conjuntos borrosos es por medio de intervalos de confianza. Esta representación se basa en las similitudes que existe entre probabilidad y lógica borrosa, aunque la probabilidad es diferente a la posibilidad de la lógica borrosa. Los intervalos de confianza son de la forma de $A = [a_1, a_2]$ si $A, B \in R$ y son intervalos de confianza, entonces es posible aplicar las operaciones de suma, sustracción, imagen, multiplicación, división e inverso. Para la suma, multiplicación, sustracción y división de $A = [a_1, a_2]$ y $B = [b_1, b_2]$, para $x \in [a_1, a_2]$, $y \in [b_1, b_2]$ se realiza punto a punto. La imagen de un intervalo es si $x \in [a_1, a_2]$, entonces $-x \in [-a_2, -a_1]$. Por último, el inverso si $x \in [a_1, a_2] \subset R_0^+$ entonces $1/x = [1/a_2, 1/a_1]$ y $A^{-1} = [a_1, a_2]^{-1} = [1/a_2, 1/a_1]$. Un intervalo de confianza puede reducir la incertidumbre al utilizar los límites inferior y superior. La información que se tenga puede ser objetiva o subjetiva. Relacionamos un concepto con el intervalo de confianza pero con un nivel de presunción (Kaufmann y Gupta, 1985)

3.1.3 Operaciones con conjuntos borrosos

Las operaciones básicas que se pueden realizar con conjuntos borrosos son la intersección, la unión y el complemento. La intersección toma el mínimo valor de los valores de pertenencia, matemáticamente se muestra en la expresión [3.1] y [3.2]. La unión se describe en las expresiones [3.3] y [3.4], donde se observa que el valor de pertenencia resultado de una unión es el máximo valor. El complemento de un valor de pertenencia está expresado en [3.5], los valores de $\mu_{\tilde{A}}, \mu_{\tilde{B}}$ y $\mu_{\tilde{C}}$ son funciones de pertenencia y \tilde{A}, \tilde{B} y \tilde{C} son conjuntos borrosos (Zimmermann, 1991).

$$\mu_{\tilde{C}}(x) \quad \tilde{C} = \tilde{A} \cap \tilde{B} \quad [3.1]$$

$$\mu_{\tilde{C}}(x) = \min\{\mu_{\tilde{A}}(x), \mu_{\tilde{B}}(x)\} \quad x \in X \quad [3.2]$$

$$\mu_{\tilde{C}}(x) \quad \tilde{C} = \tilde{A} \cup \tilde{B} \quad [3.3]$$

$$\mu_{\tilde{A}}(x) = \max(\mu_{\tilde{A}_1}(x), \mu_{\tilde{A}_2}(x)) \quad x \in X \quad [3.4]$$

$$\mu_{\tilde{A}^c}(x) = 1 - \mu_{\tilde{A}}(x) \quad x \in X \quad [3.5]$$

Sin embargo, la interpretación de la intersección y de la unión pueden tener diferentes formas. El principio de extensión permite esta interpretación. Este principio fue implementado por primera vez por Zadeh en 1965, usado para generalizar los conceptos matemáticos de los conjuntos clásicos a los conjuntos borrosos. La definición del principio de extensión dice: Si X es un producto cartesiano del universo de $X = X_1 \times X_2 \times \dots \times X_k$ y $\tilde{A}_1, \dots, \tilde{A}_k$ son conjuntos borrosos en $X = X_1 \times \dots \times X_k$, f es un mapeo de X en el universo Y , $y = f(x_1, \dots, x_k)$. Entonces el principio de extensión permite definir un conjunto borroso \tilde{B} en Y por la expresión [3.6] (Zimmermann, 1991):

$$\tilde{B} = \{y; \mu_{\tilde{B}}(y) \mid y = f(x_1, \dots, x_k), (x_1, \dots, x_k) \in X\} \quad [3.6]$$

donde

$$\mu_{\tilde{B}} = \begin{cases} \sup_{(x_1, \dots, x_k) \in f^{-1}(y)} \min\{\mu_{\tilde{A}_1}(x_1), \dots, \mu_{\tilde{A}_k}(x_k)\} & \text{Si } f^{-1}(y) \neq \emptyset \\ 0 & \text{De otro modo} \end{cases} \quad [3.7]$$

3.1.4 Aplicaciones de la lógica borrosa

Cuando se utiliza lógica borrosa en sistemas de inferencia se deben considerar varios aspectos como son: la identificación del número y parámetros de las funciones de pertenencia de los conjuntos borrosos en el antecedente, cruciales en los modelos borrosos ya que determinan el número de reglas y parámetros de las funciones de pertenencia en el consecuente. La determinación de parámetros es un proceso fuera de línea. Otros tópicos importantes son la adquisición y la representación del conocimiento (Hans, 1999; Linkens y Okola, 1993). También es importante considerar cuando se realiza un experimento para obtener información que se asegure la identificación de la causalidad, donde la información generada sea suficiente para representar todos los estados del sistema, que el experimento se pueda repetir y que exista una separación entre la entrada y la salida (Emami, Goldenberg y Türkensen, 2000). Por último, en muchas aplicaciones existe un compromiso entre rendimiento y complejidad ambos objetivos están en conflicto (Nelles, 2001).

El uso de lógica borrosa es considerado por sus resultados en un amplio rango de aplicaciones (Emami, Goldenberg y Türkensen, 2000). Varios ejemplos de esta aplicación se presentan en la literatura como son la detección de fallas, la conversión en procesos de reacciones químicas, los pronósticos de consumos de energía eléctrica, las aplicaciones en control, las herramientas de extracción, técnica de búsqueda y el reconocimientos de patrones. Las técnicas no tradicionales han

generado una gran expectativa, sobre todo la implementación híbrida de técnicas. Tres técnicas no convencionales se pueden mencionar: las redes de neuronas artificiales, los sistemas borrosos de inferencia y los algoritmos genéticos los cuales constituyen tres estudios que independientemente o mezclados son muy importantes para el modelado por computadoras (Kuo, 2000). Una de las técnicas que recientemente ha generado interés son los algoritmos genéticos como método de entrenamiento de la red de neuronas artificiales que estima los parámetros de los sistemas borrosos. El entrenamiento híbrido (mínimos cuadrados y retro-propagación) da resultados más precisos y en menor tiempo que utilizando solamente entrenamiento con retro-propagación (Dash, Liew, Rahman y Dash, 1985; Linkens y Okola, 1993). Las redes de neuronas artificiales muestran resultados importantes en la predicción de sistemas complejos no lineales mediante la aproximación a una función utilizando su proceso en paralelo. La lógica borrosa aprovecha la información en forma lingüística de los expertos además de la información puramente numérica. La combinación de las dos metodologías generan un pronóstico cuya dinámica se adapta a los cambios de ambiente, tolerancia a ruido y descubre nuevas relaciones entre las variables (Srinivasan, Liew y Chang, 1994; Linkens y Okola, 1993). La unión de técnicas no solo aprovecha sus similitudes sino existe un efecto sinérgico entre ambas (Linkens y Okola, 1993). Ejemplos de aplicaciones híbridas son: El método RIA (rule induction algorithm) de Quiang y Chouchoulas 2000, DIFNN (dual input fuzzy neural network) bajo el modelo de Mamdani y ANFIS (adaptative neural fuzzy inference system) bajo el modelo de TS (Takagi-Sugeno) propuesto por Jang (1993), combinaciones de redes y lógica borrosa (Dash, Liew, Rahman y Dash, 1985; Linkens y Okola, 1993), combinación con algoritmos genéticos (Nelles, 2001; Morimoto, Susuki y Hashimoto, 1997) y con grafos (Khoo, Ang y Zhang, 2000).

3.1.4.1 Modelado borroso

Se busca una metodología semántica para la síntesis y análisis de controles borrosos para sistemas dinámicos. Los modelos borrosos son construídos sistemáticamente a partir de datos de entrada - salida. Para tales fines, los dos modelos borrosos de inferencia más comúnmente utilizados son el de Mamdani y el de Takagi-Sugeno (Hans, 1999; Bahat, Inbar y Schenieder, 2000). El modelado de sistemas por medio de lógica borrosa implica la partición de la información donde la habilidad de la predicción está en función del conjunto de reglas y los mecanismos de razonamiento determinados (Emami, Goldenberg y Türkensen, 2000). La definición de modelos borrosos puede ser problemática de realizar y tener altos costos de tiempo, cuando se trata de minimizar la diferencia

entre los valores estimados y los valores esperados. Para minimizar el error se utiliza la suma del error cuadrático bajo modelo de Takagi-Sugeno (Morimoto, Susuki y Hashimoto, 1997).

Los modelos de un sistema borroso pueden incluir varios pasos. El primero es la encapsulación del conocimiento por medio de reglas borrosas las cuales son extraídas de los datos de entrada - salida. El segundo paso es el diseño de las reglas para cada estado del sistema. Por último, probar la estabilidad de la estructura (Emami, Goldenberg y Türkensen, 2000; Srinovasan, Liew y Chang, 1994). Otra forma de modelado borroso considera dos pasos: la configuración del modelo y la identificación del modelo. La configuración del modelo está dada por consideraciones preliminares que definen la estructura básica del modelo y los operadores borrosos apropiados (Hans, 1999). Una tercera forma propone tres pasos. El primero es la identificación donde se buscan los valores óptimos y acciones que realicen el control similar a la acción humana que crea modelos mentales en el cerebro como resultado del aprendizaje y de la experiencia. El segundo paso consiste en la búsqueda de la mejor estrategia para adquirir experiencia para realizar predicciones. Por último, se busca hacer mejoras y optimizar los sistemas complejos (Morimoto, Susuki y Hashimoto, 1997). Los modelos pueden evaluarse con varias medidas; los errores más comúnmente utilizados son el error absoluto, el error relativo, el error del rastro promedio y el error porcentual promedio (Lias y Lin, 2000).

Un modelo con lógica borrosa puede no siempre funcionar en la forma deseada, por lo que su validez es probada pragmáticamente (Babuška, Verbrugen y Can, 1999). Los modelos borrosos tratan de resolver el problema del incremento de complejidad al dividir en grupos las variables de entrada (Nelles, 2001). También los sistemas MIMO (multiple-input multiple-output) pueden considerarse como una colección o se pueden representar como un conjunto de sistemas MISO (multiple-input simple-output) (Hans, 1999; Emami, Goldenberg y Türkensen, 2000; Bahat, Inbar y Schenieder, 2000).

3.1.4.1.1 Modelo de Takagi y Sugeno

El primer aspecto a considerar es la identificación de sistemas por Takagi y Sugeno (IS). El método de IS es uno de los principales métodos en la teoría de sistemas el cual persigue desarrollar una herramienta matemática para construir modelos de sistemas. Las implicaciones contienen variables borrosas con funciones de pertenencia uni-modales las cuales son lingüísticamente entendibles. Debido a ello se llaman variables lingüísticas. Se busca tener el menor número de implicaciones; se recomienda no tener más de 5 variables en cada dimensión del espacio de entrada. Los sistemas de

entrada y salida buscan simplicidad y generalidad producto de sus herramientas matemáticas para expresar al sistema y la identificación que realice (Takagi y Sugeno, 1985).

El método borroso por Takagi y Sugeno es muy simple basado en la partición del espacio de entrada, donde para cada sub-espacio, una relación entrada-salida es formada. El razonamiento está dado por la agregación de valores inferidos por alguna implicación a una entrada. El diseño de controladores se ha tratado de realizar por la imitación de un experto humano, pero existen varios problemas como son la dificultad de poder convertir la acción humana a la acción de la máquina. Por otro lado, se pueden generar conjuntos de reglas muy grandes y no siempre el operador humano realiza el trabajo con máxima eficiencia (Takagi y Sugeno, 1985).

Cuando se habla del formato de las implicaciones borrosas y el algoritmo de razonamiento en el modelo *TS* se considera a A como un conjunto borroso, $A(x)$ es la función de pertenencia de $x \in X$. Los valores de verdad de una proposición se expresan como $|x \text{ es } A, y \text{ es } B| = A(x) \wedge B(x)$ y las implicaciones tienen la forma [3.8]

$$R: \text{if } (x_1 \text{ es } A_1, \dots, x_k \text{ es } A_k) \text{ then } y = g(x_1, \dots, x_k) \quad [3.8]$$

donde y es la variable del consecuente cuyo valor es inferido, x_1, \dots, x_k son variables de las premisas, también utilizadas en el consecuente, A_1, \dots, A_k corresponden a las funciones de pertenencia que representan un sub-espacio borroso en el cual la implicación R puede ser aplicada en el razonamiento, f es la función lógica que conecta las proposiciones de las premisas y g es la función lógica que calcula el consecuente. Pueden existir premisas no condicionadas, las cuales no estarán presentes en las premisas pero sí en el consecuente (Takagi y Sugeno, 1985; Babuška, Verbruggen y Can, 1999).

En el algoritmo de razonamiento suponemos que tenemos implicaciones de la forma $R^i (i=1, 2, 3, \dots, n)$ con (x_1^0, \dots, x_k^0) , realizándose los siguientes pasos:

- 1) Para cada implicación R^i hay una y^i que es calculada por una función g^i en el consecuente, como se muestra en la expresión [3.9], donde x representa las variables de entrada, p los coeficientes de la función g , que es un polinomio de grado 1 ó 0.

$$y^i = g^i(x_1^0, \dots, x_k^0) = p_0 + p_1 x_1^0 + \dots + p_k x_k^0 \quad [3.9]$$

- 2) El valor de la proposición $y = y^i$ es calculado por la expresión [3.10] y [3.11].

$$| y = y' | = | (x_i^0 \text{ es } A_i' \text{ and } \dots \text{ and } x_k^0 \text{ es } A_k') | \quad [3.10]$$

$$| y = y' | = | (A_i' (x_i^0) \wedge \dots \wedge A_k' (x_k^0)) | \quad [3.11]$$

- 3) La salida final y es inferida de n implicaciones y está dada por un promedio de todas las y' con sus respectivos pesos $|y=y'|$, utilizando la fórmula [3.12]

$$y = \frac{\sum |y=y'| y'}{\sum |y=y'|} \quad [3.12]$$

La identificación requiere determinar tres conjuntos de términos: x_1, \dots, x_k correspondientes a las variables que componen las premisas de implicación, A_1, \dots, A_k son las funciones de pertenencia de los conjuntos borrosos y P_0, \dots, P_k son los parámetros en el consecuente. Cabe hacer notar que no todas las variables están en las premisas y hay una jerarquía para la determinación de los mismos. Esta jerarquía determina o selecciona las variables de las premisas, después se determinan los parámetros en las premisas y por último se determinan los parámetros en el consecuente (Takagi y Sugeno, 1985). En el modelo borroso propuesto por Takagi-Sugeno, el consecuente no contiene conjuntos borrosos sino polinomios, siendo los más populares para describir series de tiempo los polinomios lineales ya que pueden interpretarse fácilmente como modelos lineales. Para un modelo de Takagi-Sugeno la concretización se debe interpretar como un promedio ponderado por pesos. El resultado del polinomio ($f_i(x)$) que tiene un grado de pertenencia a la regla ($\mu_i(x)$) es calculado sobre la base del antecedente, por lo que el valor estimado se obtiene con la ecuación [3.13], donde M es el número de reglas y $y(x)$ es el valor no borroso estimado del valor real. La interpretación en términos de reglas es la mayor diferencia que existe entre sistemas borrosos y otras aproximaciones lineales. Los factores que influyen en la interpretación de un sistema borroso son el número de reglas, el número de antecedentes en las reglas, dimensión del conjunto borroso de entrada, orden de los conjuntos borrosos, y la forma de normalización de las funciones de pertenencia (Nelles, 2001).

$$y(x) = \sum_i^M \mu_i(x) f_i(x) \quad [3.13]$$

3.2 Redes neuronales

3.2.1 Introducción a las redes neuronales

La historia de las redes de neuronas artificiales comienza con McCulloch y Pitts en 1943 y Hebb en 1949, los cuales construyeron los primeros modelos con actividad neurobiológica y aprendizaje, respectivamente. Rosenblatt en 1962 verificó la habilidad computacional de los perceptrones de una sola capa. El análisis determinó que no podían aprender más que la clase de funciones lineales. Rumelhart, Hinton y Williams en 1986 retomaron el interés por las redes neuronales al proponer el método de retro-propagación para entrenamiento (Plummer, 2000). La teoría del modelo de las redes neuronales artificiales es un modelo matemático que está inspirado en la estructura y funcionamiento de los sistemas nerviosos biológicos (Hilera y Martínez, 2000; Plummer, 2000; Hecht-Niesen, 1988). Las redes neuronales sirven para procesar información por medio de su estado dinámico como respuesta a entradas, en las cuales se realizan dos funciones: 1) la agregación de las entradas desde otras neuronas o el ambiente externo y 2) la generación de una salida con la agregación de las entradas a través de la función de transferencia. La salida es afectada por el peso de la liga entre neuronas (Linkens y Okola, 1993). Las redes neuronales artificiales realizan un proceso de aprendizaje para cumplir con sus funciones (Plummer, 2000).

3.2.2 Estructura y características de las redes neuronales

Las redes de neuronas artificiales están formadas de diferentes unidades interconectadas llamadas *neuronas*, las cuales representan la unidad fundamental (Hilera y Martínez, 2000; Plummer, 2000; Hecht-Niesen, 1988; Kolarik y Rudorfer, 1974). La estructura de una neurona está formada por el cuerpo, una rama principal conocida como axón, varias ramas más pequeñas conocidas como dendritas y las sinapsis, que son conexiones entre neuronas. La característica distintiva de estas células es su capacidad de comunicarse. Un modelo de estas conexiones se muestra en la Figura 3.2. Las neuronas están agrupadas por capas, la primera capa se le conoce como capa de entrada y la última capa se le conoce como capa de salida, todas las demás capas son ocultas (Kolarik y Rudorfer, 1974).

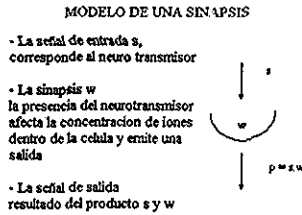


Figura 3.2: Modelo de una sinapsis.

La micro-estructura de una red está definida en razón del número de neuronas, el estado de activación de cada neurona, el patrón de conectividad, la regla de propagación, la función de transferencia, la regla de activación y las reglas de aprendizaje. Una red neuronal es un modelo estadístico no paramétrico para la extracción de relaciones no lineales de los datos (Hilera y Martínez, 2000).

Los estados del sistema en un tiempo t se especifican por un vector de N números reales, uno por cada neurona de la red. A dicho vector se le conoce como *estado de activación*. El procesamiento que realiza la red es la evolución de un patrón de activación de la red que se compone a través del tiempo. Todas las neuronas de la red se hallan en cierto estado; en forma simplificada se dice que existen los estados de reposo y excitado. Los valores de activación pueden ser discretos o continuos. Las diferentes neuronas se unen entre ellas por conexiones. Cada neurona transmite señales de su salida a las entradas de otras neuronas. Asociada a cada neurona existe una *función de salida*, también llamada *función de transferencia* que transforma el estado actual de activación $a_i(t)$ en una señal de salida $y_i(t)$, es decir $y_i(t) = f(a_i(t))$. Existen diferentes tipos de funciones de transferencia tales como la función identidad, la función escalón, la función lineal - mixta, la función sigmoideal y la función gaussiana. La función sinoidal se aplica cuando la función es continua, mientras que si es discreta se utiliza la función escalón y suelen aplicarse el mismo tipo a todas las neuronas. Las redes de neuronas artificiales son clasificadas sobre la base de la naturaleza de las funciones de transferencia de los nodos (Linkens y Okola, 1993). Existe un modelo que proporciona una suma de pesos de los vectores de entrada efectuando solo un mapeo lineal. Adiciona no-linealidad al ejecutar funciones más complejas sobre el vector de entrada y desempeña funcionalidad no lineal sobre la activación de la red. Si bien es difícil explicar como funciona se contraponen a lo bien que funcionan (Plummer, 2000).

Las conexiones que unen las neuronas tienen asociado un *peso* el cual adquiere el conocimiento. Cada sinapsis entre la neurona i y la neurona j está ponderada por un peso w_{ij} . Normalmente se considera que el efecto de cada señal es aditivo, de tal forma que la entrada que recibe una neurona

net_j (potencial postsináptico) es la suma del producto de cada señal de entrada (y_i) por el valor de la sinapsis (peso w_{ij}) que conecta ambas neuronas donde N es el número de entradas (Hilera y Martínez, 2000; Kolarik y Rudorfer, 1974), como lo muestra la ecuación [3.14]

$$net_j = \sum_i^N w_{ij} * y_i \quad [3.14]$$

La *regla de propagación* es el procedimiento a seguir para combinar los valores de entrada a una unidad con los pesos de las conexiones que llegan a dicha unidad. Los valores de entrada son escalados a valores en el rango de [0,1] y son usados como los *niveles de activación* de la capa de entrada. En ruta hacia adelante los valores de entrada son propagados hacia las capas ocultas y de ahí hacia la capa de salida (Kolarik y Rudorfer, 1974) Para cambiar el estado de una neurona cuando es estimulada su entrada, se requiere una función o regla de activación. La función F debe combinar el valor de sus entradas como potencial postsináptico con el estado actual de la neurona. Esto es, $a(t+1) = F(a(t), net_i)$ La función de activación más común es la función identidad donde el valor en el siguiente estado corresponde con el valor del potencial postsináptico (Kolarik y Rudorfer, 1974). Normalmente la función de activación no está centrada en el origen, sino que existe cierto desplazamiento debido a las características internas propias de cada neurona y la variación de una a otra. Este valor se le conoce como *umbral de activación* de la neurona i y se denota como θ_i (Hilera y Martínez, 2000) En la Figura 3.3 se observan los elementos básicos de una neurona.

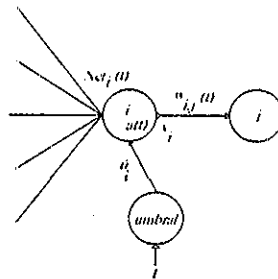


Figura 3.3: Elementos básicos de una neurona artificial.

La *topología, arquitectura o meso estructura* de las redes de neuronas artificiales consiste en la organización y disposición de las neuronas en la red formando capas o agrupaciones de neuronas alejadas de la entrada y salida de la red. En este sentido, los parámetros fundamentales de la red son: el número de capas, el número de unidades por capa, el grado de conectividad y el tipo de conexiones entre neuronas (Kolarik y Rudorfer, 1974) La arquitectura apropiada es un elemento

crítico en el funcionamiento de la red, la cual es determinada por el número de entradas de la serie de tiempo, número de neuronas ocultas, intervalo inicial de los pesos aleatorios, de los parámetros de tasa de aprendizaje como también el momento y el número de patrones seleccionados para evitar sobre-entrenamiento. Si la red es muy pequeña no aprende todas las relaciones y si es muy grande, puede capturar los efectos del ruido (Thiesing y Vornberger, 1997; Tang, Hsich, Tang y Haimes, 2001; Kolarik y Rudorfer, 1974).

3.2.3 Aprendizaje de las redes neuronales

El *aprendizaje* es la modificación del comportamiento inducido por la interacción con el entorno y como resultado de experiencias conducentes al establecimiento de nuevos modelos de respuestas a estímulos externos (Hilera y Martínez, 2000). Este consiste en operar en un espacio vectorial el cual realiza un mapeo sobre el espacio vectorial del problema (Lawrence, Tsoi y Giles, 96). En el caso de las neuronas naturales, el conocimiento se encuentra almacenado en las sinapsis mientras que en las neuronas artificiales el conocimiento se encuentra almacenado por los pesos de las conexiones, por lo cual, todo proceso de aprendizaje se relaciona con los cambios de los pesos de las conexiones. Durante el aprendizaje se modifican los pesos en respuesta a una información de entrada. Los cambios que se producen durante el proceso de aprendizaje natural se reducen a la destrucción, modificación y creación de conexiones entre neuronas (Hilera y Martínez, 2000). Las formas de diseño de realización del aprendizaje son por aprendizaje global, cuando los cambios en los pesos de las conexiones afectan todas las salidas; mientras que para un diseño local, el cambio de pesos solo afecta una parte de la red. Un entrenamiento global se realiza fuera de línea mientras que las redes con entrenamiento local se realizan en tiempo real, en línea (Linkens y Okola, 1993). La finalidad de las redes con alimentación hacia adelante, multi-capas y con entrenamiento supervisado es generar un mapeo entre los valores de entrada y los valores de salida (Thiesing y Vornberger, 1997). Dos parámetros para el entrenamiento de la red de neuronas son la tasa de aprendizaje y el momento. El método de retropropagación es un método iterativo que por un gradiente se aproxima a los pesos óptimos, siendo la *tasa de aprendizaje* una constante que controla el tamaño del salto en cada iteración, como un porcentaje del gradiente. El *momento* consiste en sumar una fracción del cambio anterior cuando se calcula el valor del cambio actual. Estos parámetros son determinados dependiendo de las características de los fenómenos (Tang, Hsich, Tang y Haimes, 2001; Kolarik y Rudorfer, 1974).

El proceso llamado entrenamiento minimiza el error. La medición del error es realizada con varias medidas como son el error cuadrático medio (S_m) entre los valores estimados (\hat{X}) y los valores

esperados (X), como se expresa en la ecuación [3.15] para n datos. El proceso de aprendizaje se considera que ha finalizado cuando los pesos de las conexiones de las redes permanecen estables ($dw/dt \approx 0$) (Hilera y Martínez, 2000)

$$s_m = \sqrt{\frac{\sum_{i=1}^n (X_i - \hat{X}_i)^2}{n}} \quad [3.15]$$

Los criterios que determinan cómo una red aprende se conocen como *regla de aprendizaje* de la red. De forma general se pueden considerar dos tipos de reglas: las que responden a lo que habitualmente se le conoce como aprendizaje supervisado y los que corresponden a un aprendizaje no supervisado. La diferencia fundamental entre ambos tipos estriba en la existencia o no de un agente externo, conocido como supervisor que controle el proceso de aprendizaje de la red (Hilera y Martínez, 2000).

3.2.4 Método de retro propagación

Uno de los métodos de entrenamiento más utilizado es el método de retro-propagación donde se presentan pares del vector de entrada y el vector de salida en la entrada, las entradas se propagan a través de todas las capas ocultas hasta llegar a la salida. El siguiente paso es comparar la salida con los valores deseados. El error obtenido es propagado en sentido inverso modificando los pesos de la red (Linkens y Okola, 1993; Kolarik y Rudorfer, 1974; Thiesing y Vornberger, 1997). A esta fase se le denomina de entrenamiento. La otra etapa, es la de prueba la cual evalúa la capacidad de generalización de la red. La *generalización* es la capacidad de estimar con datos no entrenados y el *sobreentrenamiento* es cuando los resultados tienen bajos errores por que ajustan mejor los pesos para estimar las observaciones de entrenamiento, pero no aprenden sobre el fenómeno (Plummer, 2000; Dash, Liew, Rahman y Dash, 1985)

3.2.5 Aplicaciones con redes neuronales

Las redes neuronales tienen la capacidad de *aprendizaje adaptativo*, la *autoorganización* y la *tolerancia a fallas* (Kuo, 2000; Hilera y Martínez, 2000). La aplicación de redes de neuronas artificiales se justifica más cuando los fenómenos tienen un pequeño número de observaciones, existen variables que no se pueden medir, con la presencia de ruido, no estacionarios, con relaciones no lineales de alta complejidad y poco entendidos (Lawrence. Isoi y Giles, 96; Kolarik y Rudorfer, 1974; Moody, 1995; Yang, Chen y Wang, 2000; Plummer, 2000; Molina. Isasi, Berlanga y Sanchis,

2000). Existen varias aplicaciones de las redes neuronales en la biología, industria, finanzas, medio ambiente, medicina y otras actividades humanas (Hilera y Martínez, 2000; Plummer, 2000; Moody, 1995; Morimoto, Susuki y Hashimoto, 1997; Yu, Gomma y Williams, 2000; Kolarik y Rudorfer, 1974; Lawrence Isoi y Giles, 96)

Un problema central en la ciencia es la predicción de secuencias temporales futuras, tales como el tiempo y tasas de cambio, entre otras (Thiesing y Vornberger, 1997; Babuška, Verbrugen y Can, 1999) Las consideraciones específicas del fenómeno son importantes para mejorar las predicciones pero no son absolutamente necesarias cuando se utilizan redes neuronales (Tang, Hsich, Tang y Haimes, 2001). Las redes de neuronas artificiales producen una inmediata decisión con un mínimo de computación con buena precisión (Srinivasan, Liew y Chang, 1994)

Existen aplicaciones prácticas de las redes neuronales como son la supervisión del funcionamiento de equipo (Molina, Isasi, Berlanga y Sanchis, 2000; Yang, Chen y Wang, 2000) y el pronóstico del fenómeno del Niño en el sur del pacífico (Tang, Hsich, Tang y Haimes, 2001)

3.2.6 Redes neuronales evolucionadas (EANN)

Una red adaptativa es el término genérico aplicado a las redes de neuronas artificiales con alimentación hacia adelante con capacidad de aprendizaje supervisado. Todos los nodos de una red adaptativa son adaptables lo cual significa que las salidas dependen de los parámetros pertenecientes a cada nodo y las reglas específicas de cómo estos parámetros pueden ser modificados para minimizar el error (Jang, 1993)

Las redes neuronales evolucionadas son redes cuya fase de entrenamiento es resultado de un método de la computación evolutiva (EANN *evolutionary artificial neural network*) El tipo de entrenamiento puede ser visto desde tres diferentes niveles, que son los pesos de las conexiones, la arquitectura de la red y las reglas de aprendizaje (Yao, 1999) Existen dos posibles tendencias para el entrenamiento evolutivo, por medio de algoritmos genéticos o bien por medio de programación genética

La evolución con algoritmos genéticos realiza los pasos siguientes (Yao, 1999) El primer paso es decodificar cada individuo (genotipo) en la actual generación dentro de un conjunto de conexiones de peso y construir la correspondiente ANN con los pesos El siguiente paso evalúa cada ANN, calcula el error cuadrático medio entre los valores estimados por la red y los valores reales. La aptitud o *fitness* de un individuo es determinado por este error. Entre mayor sea el error éste deberá tener una menor aptitud. El paso 3 consiste en la selección de los padres de la siguiente generación a través de su aptitud. por último, se aplican los operadores genéticos tales como cruzamiento y

mutación a los padres para obtener la siguiente generación. El aspecto más importante para la aplicación del algoritmo genético radica en la representación. La representación consiste en presentar cada conexión de la red con un gen (Yao, 1999)

Por medio de la programación genética se evoluciona la estructura de la red neuronal considerando los siguientes aspectos: número de capas, número de elementos en cada etapa, conectividad de la red, función de activación, paradigma de entrenamiento, entradas, salidas, casos de entrenamiento a ser utilizados, medición del error, definición de parámetros del control del entrenamiento y definición de criterios de terminación. En realidad todas esas variables pueden ser introducidas a un proceso evolutivo para determinar la red más adecuada. Cuando se utiliza la programación genética se definen cuatro tipos de elementos para formar las expresiones. Las expresiones en PG son árboles que tienen nodos función y terminales. Los nodos que simbolizan el proceso de una neurona (P), los nodos que representan el peso de la función (W), los nodos que contienen entrada de datos (D) y los nodos que representan una función aritmética o una constante aleatoria (F). La forma de construcción de los árboles es la siguiente: la raíz del árbol debe ser siempre un nodo P . Los siguientes niveles pueden aceptar cualquier tipo de nodo. Los nodos tipo W y las constantes aleatorias se consideran como nodos hojas; éstos es, pertenecientes al conjunto de terminales. Solo se permiten nodos W al lado de un nodo D . Las funciones aritméticas solo permiten como parámetros constantes aleatorias, o el resultado de otra función aritmética. Estas reglas se aplican recursivamente (Koza, 1992).

3.2.7 Redes tipo ART

La red ART fue propuesta por Carpenter y Grossberg durante los años de 1986 a 1987. Estos autores han realizado una modificación a su red inicial a la cual llamaron ART2, la cual es una versión que utiliza valores reales como entrada. Si bien la arquitectura es semejante a la versión inicial, varían en que los pesos de las conexiones hacia adelante y hacia atrás son iguales. La arquitectura de una red ART básicamente consta de capas de nodos entre las cuales existen conexiones hacia delante (feedforward) y hacia atrás (feedback). Pueden existir otros elementos como las estructuras conocidas con el nombre de sub-sistema de atención y sub-sistema de orientación. El sub-sistema de atención tiene como función el reconocimiento y clasificación de la información aprendida. El sub-sistema de orientación utiliza el parámetro de vigilancia para reiniciar las salidas cuando no existe un patrón válido. La arquitectura más simple de la red ART se presenta en la Figura 3.4 donde e_i representa las neuronas de entrada, s_i representa la i -ésima salida, w_i las conexiones hacia delante y v_i son las conexiones hacia atrás. Una red ART es un grafo

bipartita completo en sus conexiones hacia adelante y completo en sus conexiones hacia atrás (Molina, Isasi, Berlanga y Sanchis, 2000; Hilera y Martínez, 2000).

Las redes *ART* no utilizan un agente externo para medir el desempeño de la red sino que utilizan los datos de entrada. Las redes *ART* son modelos de aprendizaje competitivo, es un flujo de patrones de entrada a la red, donde los pesos se adaptan en forma de memoria de largo tiempo (*LTM*). Los patrones de entrada, como en otras redes, son normalizados. La respuesta es la neurona ganadora. En términos de percepción, los patrones de entrada son clasificados dentro de categorías de reconocimiento mutuamente exclusivas. La distribución de probabilidad con la que los patrones se presentan en la entrada y la secuencia de presentación es importante para la estabilidad de la red. Sin embargo, la inestabilidad no es un problema que se presente normalmente en redes con aprendizaje competitivo. Una de las propiedades de las redes *ART* es su capacidad para el aprendizaje de acuerdo a su experiencia y su habilidad para codificar en forma comprimida en la representación interna. (Carpenter y Grossberg, 1988).

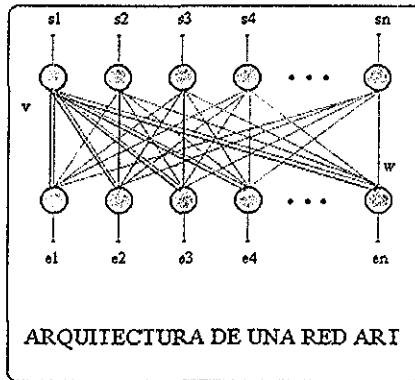


Figura 3.4: Arquitectura de una red ART.

El funcionamiento de la red *ART2* es el siguiente: Se presenta una información de entrada a la red por el vector E_k [3 16] formado de las entradas $e_i^{(k)}$. Cada neurona de entrada (n_{ei}) recibe su valor correspondiente y los transmite por todas las conexiones hacia adelante. Después, cada neurona de salida (n_{sj}) calcula todos sus valores de salida (S_{nsj}) por medio de la ecuación [3 17], donde ϵ es un valor menor de $1/M$, siendo M el número de neuronas, N el número de aristas que inciden, f la función de transferencia en forma de escalón y w_{ji} los pesos de las conexiones hacia adelante. Una vez que se tiene un valor de salida, se compete con todas las neuronas de salida hasta que solo exista una ganadora. La neurona ganadora tiene una salida 1 y las otras 0, siendo k -ésima entrada, j -ésima neurona de salida, i -ésima neurona entrada.

$$E_k = (e_1^{(k)}, \dots, e_n^{(k)}) \quad [3.16]$$

$$S_{ns_j}(t+1) = f \left[S_{ns_j}(t) - \varepsilon \sum_{\substack{p=1 \\ p \neq j}}^M S_{ns_p}(t) + \sum_{i=1}^M w_{ji} S_{nc_i}(t) \right] \quad [3.17]$$

La neurona vencedora ($S_{ns_j}^*$) envía su salida a través de las conexiones hacia atrás (v_{ij}^*), cada i -ésima neurona de la capa de entrada recibe el valor, como lo muestra la ecuación [3.18]

$$x_i = \sum_{j=1}^M v_{ji} S_{ns_j} = v_{ij}^* \quad [3.18]$$

Se compara la entrada inicial con los pesos de las conexiones y se obtiene una *relación de semejanza* [3.19]. La relación de semejanza entre ambas informaciones es confrontada con el parámetro de vigilancia ρ . Si la relación de semejanza es estrictamente menor a ρ , entonces la neurona vencedora no presenta la categoría adecuada, lo cual implica la creación de una nueva categoría. Si la relación de semejanza es mayor o igual al parámetro de semejanza, se ajustan los pesos del prototipo reconocido. El ajuste de peso se realiza aplicando las ecuaciones [3.20] y [3.21], siendo $N(t)$ el número de patrones en el grupo de patrones vencedores.

$$\text{relación de semejanza} = \sum_j \text{abs}(e_{ij} - w_{ji}) \quad [3.19]$$

$$w_{ji}^{(t+1)} = \frac{(e_{ij} + w_{ji} N(t))}{N(t) + 1} \quad [3.20]$$

$$v_{ji}^{(t+1)} = w_{ji}^{(t+1)} \quad [3.21]$$

Sin embargo, las redes *ART* presentan ciertas desventajas. La más importante es su alta dependencia al parámetro de vigilancia. También puede afectar los resultados la forma en que son introducidos los datos de entrada. Las redes *ART* son muy sensibles a ruido o distorsiones. El almacenamiento no es eficiente ya que requiere de dos veces del espacio para los pesos, además que son los mismos. Un aspecto relevante es que puede existir una degradación de los prototipos, en específico para redes *ART1* por la naturaleza destructiva de la función AND (Molina, Isasi, Berlanga y Sanchis, 2000; Hilera y Martínez, 2000). A pesar de estas desventajas, las redes *ART* pueden tener un buen desempeño. Las redes *ART* son estructuras simples, solo tienen dos capas, una neurona por cada entrada y una neurona para cada conjunto de patrones. Su aprendizaje es completamente en línea, no requiere de lapsos de entrenamiento. La principal ventaja de las redes *ART* es no perder lo aprendido cuando llegan nuevos vectores de entrada.

3.3 Computación evolutiva

La computación evolutiva es un conjunto de algoritmos de adaptación inspirados en los procesos de la evolución y la selección natural, los cuales mantienen una población de individuos que evolucionan de acuerdo a reglas de selección y de otros operadores genéticos como son el cruzamiento y la mutación. De la teoría de la evolución propuesta por Darwin, se consideraron la selección como mecanismo de evolución, la supervivencia del más apto y las mutaciones. De las leyes de Mendel se consideró la transmisión de las características de padres a hijos (Goldberg, 1989; Linkens y Okola, 1993). La computación evolutiva considera varios algoritmos que se muestran en la Figura 3.5, además de quienes propusieron la primera versión.

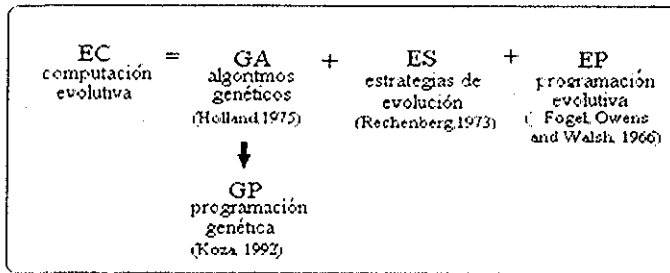


Figura 3.5: Algoritmos de la computación evolutiva.

Las dos técnicas evolutivas utilizadas en este trabajo son el algoritmo genético y la programación genética, las cuales presentan los siguientes pasos:

PROGRAMA Basado en evolución

```

=====
t = 0
Crea la población inicial P(t)
Evalúa la población inicial P(t)
Mientras (criterio de terminación) Hacer
    t = t + 1
    Selección de individuos para reproducción
        P(t) desde P(t-1)
    Alterar P(t)
    Evaluar la nueva población P(t)
terminar
=====
    
```

Todos los algoritmos de la computación evolutiva comparten la definición de una población de N individuos representados en forma binaria, real o estructuras más complejas; tienen una función de adaptabilidad para optimizar o evaluar cada individuo, mecanismos de selección del más apto, mecanismos de sustitución para mantener la población constante y operadores basados en procesos naturales. También mantienen términos basados en los conceptos biológicos; por ejemplo, individuo es una solución, población un conjunto de soluciones, adaptabilidad representa la calidad de la solución, cromosoma es la representación de una solución, gen corresponde a una parte de una solución, recombinación-mutación son operadores de búsqueda y la selección natural representa la re-utilización de soluciones sub-óptimas.

Actualmente existen tres tipos de métodos de búsqueda que pueden ser considerados como tradicionales. Estos son, los métodos basados en el cálculo, los métodos enumerativos y los métodos totalmente aleatorios. Sin embargo, dichos métodos son poco robustos más no inútiles. Los métodos tradicionales tienden a realizar una búsqueda local, a veces dependientes de las propiedades de las funciones tal que sean derivables en todos sus puntos, continuas, unimodales y en ocasiones requieren de información auxiliar para funcionar (Goldberg, 1989).

3.3.1 Algoritmo genético

El algoritmo genético fue diseñado en su primera versión por Holland en 1975. Es un algoritmo de búsqueda basado en mecanismos de la selección natural, donde se presenta una eficiente explotación de la información histórica para especular cual puede ser el siguiente espacio de búsqueda que mejore el desempeño. Los objetivos iniciales eran realizar una rigurosa abstracción de los procesos naturales de adaptación y diseñar sistemas artificiales basados en mecanismos de sistemas naturales. Existe una terminología donde hay un símil entre los conceptos biológicos y los conceptos en el algoritmo. La Tabla 3.1 muestra unos ejemplos de dichos términos, que pueden ser válidos para otros algoritmos de la computación evolutiva. Los algoritmos genéticos también son considerados como algoritmos de optimización que incorporan conceptos biológicos dentro de sistemas analíticos. Los algoritmos genéticos pertenecen a la familia de los algoritmos que realizan optimización probabilística basada en la evolución natural. Estos algoritmos consideran poblaciones que sufren modificaciones durante varias generaciones (Nelles, 2001; Khoo, Ang y Zhang, 2000).

Tabla 3.1: Comparación de términos biológicos con términos del algoritmo.

<i>Evolución y genética natural</i>	<i>Algoritmos genéticos</i>
<i>Individuo</i>	<i>Solución a un problema</i>
<i>Población</i>	<i>Conjunto de soluciones</i>
<i>Adaptabilidad</i>	<i>Calidad de la solución</i>
<i>Cromosoma</i>	<i>Representación de la solución</i>
<i>Gen</i>	<i>Parte de la representación, por ejemplo un parámetro</i>
<i>Alelo</i>	<i>Valor específico de un gen</i>
<i>Recombinación y mutación</i>	<i>Operadores de búsqueda</i>
<i>Selección natural</i>	<i>Reutilización de soluciones</i>
<i>Fenotipo</i>	<i>Solución en términos del dominio del problema</i>
<i>Genotipo</i>	<i>Solución en términos de la representación</i>

Los algoritmos genéticos son robustos con aplicaciones en una amplia variedad de problemas. El tipo de problemas que más se aplica es donde se permite un rango de error; ésto es, donde no se requiera una solución exacta. Resuelve problemas que los métodos tradicionales no pueden o presentan dificultades, como por ejemplo, cuando las funciones tienen discontinuidades, no son derivables en todos sus puntos, existe multi-modalidad y para dominios poco estudiados.

El algoritmo genético difiere de los métodos tradicionales en los siguientes puntos (Goldberg, 1989):

- 1) Trabaja con una codificación de los parámetros no con los parámetros en sí.
- 2) Trabaja con varios puntos de solución, no sobre un solo punto
- 3) Usa la evaluación objetivo en lugar de derivadas u otros auxiliares conocidos
- 4) Usa reglas de transición probabilista no determinista que son aplicables a problemas complejos, con alta dimensión, con poca experiencia o falsos óptimos

La identificación de los individuos puede realizarse de dos maneras, fenotípicamente que representa en sí el valor del individuo y por genotipo que es la codificación de dicho fenotipo

El proceso de los algoritmos genéticos inicia con una codificación de la solución, lo cual implica un tipo de dato, un alfabeto finito y un número de cadenas a generar mediante un alfabeto finito. El tipo de dato puede ser binario o real (Goldberg, 1989).

Generalmente por medio de procesos aleatorios un algoritmo genético produce soluciones del problema dado en forma codificada. El tener las soluciones codificadas evita la dependencia de las propiedades de las funciones tal que sea derivable en todos sus puntos y continua. En los métodos tradicionales, la búsqueda se realiza por la definición del siguiente punto a buscar, conocida como la técnica punto a punto. Con la técnica punto a punto existe la posibilidad de localizar puntos óptimos de naturaleza local. Los algoritmos genéticos, al evaluar varias soluciones a la vez, permiten

búsquedas más amplias y con la aplicación de los operadores genéticos permite la búsqueda de óptimos globales. Muchos métodos requieren de una información auxiliar para resolver el problema. Los algoritmos genéticos pueden apoyarse en dicha información pero no es indispensable.

Los algoritmos genéticos utilizan la información histórica de la población de soluciones para determinar hacia donde evolucionará la población. El azar en los algoritmos genéticos se utiliza para determinar el espacio donde se realizará la búsqueda.

El algoritmo no tiene una complejidad más allá de copiar y mezclar cadenas (string), el por qué funciona tiene una explicación más sutil y poderosa.

3.3.1.1 Pasos del algoritmo genético

El algoritmo genético sigue los pasos del programa de la computación evolutiva. Crea una población aleatoriamente, evalúa de acuerdo a la adaptación a su ambiente y aplica operadores. Los operadores básicos aplicados son la reproducción, la recombinación y la mutación (Goldberg, 1989).

Los individuos son evaluados en base a una función objetivo o varias funciones, que definen al problema. La evaluación es mapeada a otro dominio que califica que tan buena o mala es la solución propuesta por el individuo al problema dado; a este proceso se le conoce como asignación de la aptitud. Los dos métodos más comunes de asignación de aptitud son el método de ranking (ordenación) y el método de escalamiento. Los valores de aptitud generalmente se miden sobre escala de valores positivos y tratan de mantener un rango adecuado entre el individuo de peor evaluación y el de mejor evaluación.

La reproducción implica un proceso de selección vía su habilidad para sobrevivir como respuesta a su capacidad de adaptación al ambiente. Existen varios métodos de selección, siendo los más comunes el basado en forma proporcional a la aptitud, como el método de la ruleta y el método estocástico universal. La ruleta define áreas proporcionales a la aptitud relativa de cada individuo. En cada giro de la ruleta se selecciona al individuo que marca la ruleta y se realizan tantos giros como individuos se requiera seleccionar. La Figura 3.6 muestra la idea del método de la ruleta.

La recombinación es un operador primario y simula a la reproducción sexual. El número de soluciones que existen en una época no son solo tantos como individuos existen sino que también hay que considerar todas las potenciales soluciones que se pueden obtener al mezclar las soluciones existentes. Esta es una de las principales razones por la cual los algoritmos genéticos no son puramente estocásticos. Una vez obtenidos los padres, la siguiente etapa del operador determina una

mezcla de los genomas paternos para producir su descendencia. Un ejemplo de la recombinación se presenta en la Figura 3.7. La mutación se considera un operador secundario y asexual, solo interviene un individuo. La mutación se considera con valores bajos como en la naturaleza. Es poco comprendido el efecto de la mutación Sin embargo, se ha establecido que su mayor aporte es incrementar la diversidad en la población. El método de mutación es descrito en la Figura 3.8.

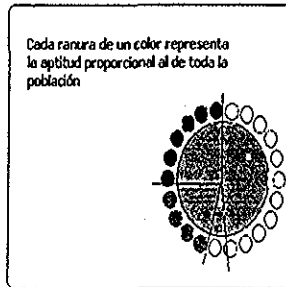


Figura 3.6: método de ruleta.

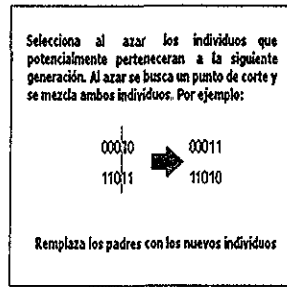


Figura 3.7: Recombinación.

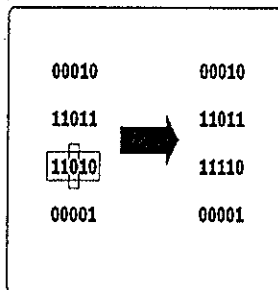


Figura 3.8: Mutación.

TESIS CON FALLA DE ORIGEN

3.3.1.2 Parámetros

Los parámetros más comunes en el algoritmo genético son el tamaño de la población, el método de selección, la tasa de reproducción, la tasa de recombinación, el tipo de recombinación, la tasa y el tipo de mutación.

3.3.1.3 Aplicación del algoritmo genético

Una de las aplicaciones más importantes de los algoritmos genéticos es en problemas de optimización, en especial multi-objetivo. Los problemas multi-objetivo se han resuelto

tradicionalmente por otras técnicas convencionales; sin embargo, éstas son poco útiles cuando la dimensión del problema es muy grande. Los algoritmos genéticos se han aplicado en el diseño que utiliza una técnica multi-objetivo basada en Pareto (Cvetkovic y Parmee, 1999). Otras aplicaciones muy comunes son en la determinación de rutas en la robótica (Pratihari, Deb y Ghosh, 1999; Kubota, Morioka y Kojima, 1999), la prueba del funcionamiento de circuitos eléctricos (Corno, Reorda y Squillero, 1999) y el diseño de circuitos eléctricos (Vassilev, Miller y Fogarty, 1999). Las aplicaciones híbridas con técnicas no convencionales es muy común; por ejemplo en sistema borroso de inferencia (Nelles, 2001). También son utilizados para encontrar las funciones de membresía y las reglas de control óptimas (Kubota, Morioka y Kojima, 1999; Morimoto, Susuki y Hashimoto, 1997). Otra aplicación híbrida se implementa con redes neuronales artificiales (Chaiyaratana y Zalzal, 1999; Zitang, Ohm y Mühlhenbein, 1997). El uso de algoritmos genéticos permite acelerar el proceso de entrenamiento y obtener una mayor precisión en los resultados (Kuo, 2000).

Aplicaciones específicas de predicción y pronóstico se han propuesto como: la predicción de la producción de reservas de petróleo (Soleng, 1999) y el pronóstico en problemas financieros (Mehta y Bhattacharyya, 1999).

3.3.1.4 Teorema del esquema

Un esquema es una plantilla que describe a un conjunto de strings (cadenas) con similitudes en ciertas partes o posiciones de las cadenas. El teorema del esquema formaliza el funcionamiento de los algoritmos genéticos, prediciendo los efectos de la selección por el método de la ruleta, mutación uniforme y recombinación en un solo punto sobre los individuos de la población.

Los individuos son cadenas de caracteres que pertenecen a un alfabeto V . Si a_i es un gen, el cromosoma de un individuo A está formado de una secuencia no necesariamente ordenada de genes. Los valores de cada gen puede cambiar durante el tiempo, el cromosoma del individuo j en el tiempo t se representa por $A_j(t)$. Un esquema puede tener todos los elementos del alfabeto V más un elemento $*$ que es un comodín el cual representa a cualquiera de los elementos de V . Una cadena con comodines se le conoce como esquema. El número de esquemas posibles es calculado con la expresión [3.22], donde k es la cardinalidad de V y l es la longitud de la cadena. El orden del esquema H , se denota por $o(H)$, representando el número de posiciones con valores fijos. La longitud de definición de un esquema H se denota por $\delta(H)$ la cual representa la distancia entre la primera posición y la última posición con valor fijo. $m(H,t)$ es el número de individuos pertenecientes al esquema H en el tiempo t . La proporción de los individuos por efecto de

reproducción para el siguiente tiempo es detallada en la expresión [3 23], donde $f(H)$ es la aptitud del esquema y en la ecuación [3 24] se muestra la aptitud promedio de la población. La probabilidad de supervivencia (P_s) de un esquema por el efecto de la recombinación está dado por la ecuación [3 25], donde P_c es el porcentaje de recombinación. La medida de la supervivencia de un esquema por efecto de la mutación está dada por la expresión [3 26], donde P_m es el porcentaje de mutación. El efecto conjunto de los tres factores puede verse en la expresión [3 27]

$$(k+1)^l \quad [3 22]$$

$$m(H, t+1) = m(H, t) \frac{f(H)}{\bar{f}} \quad [3 23]$$

$$\bar{f} = \frac{\sum f_j}{n} \quad [3 24]$$

$$P_s \geq 1 - P_c \frac{\delta(H)}{l-1} \quad [3 25]$$

$$1 - o(H) * P_m \quad [3 26]$$

$$m(H, t+1) \geq m(H, t) \frac{f(H)}{\bar{f}} \left[1 - P_c \frac{\delta(H)}{l-1} - o(H)P_m \right] \quad [3 27]$$

3.3.2 Programación genética

La programación genética fue propuesta por Koza en 1992. La programación genética se considera una variante del algoritmo genético propuesto por Holland en 1975. Esta variación se presenta sobre la base de la representación, mientras en el algoritmo genético original la representación se realiza por números binarios, la programación genética representa estructuras ejecutables en forma de árbol. Otra diferencia es que las estructuras representadas con algoritmos genéticos tienen, generalmente, dimensiones fijas. Con programación genética la dimensión es variable (Koza, 1992). La programación genética no aprovecha solamente la búsqueda estocástica para resolver el problema, también lo resuelve utilizando la información histórica y la información de todos los individuos de la población. La forma de representar la estructura jerárquica es por medio de árboles los cuales son en sí expresiones ejecutables.

El espacio de búsqueda de la programación genética está formado por la combinación de dos conjuntos: el conjunto de funciones y el conjunto de terminales. Esta combinación surge de un proceso recursivo donde cada función es un nodo interno del árbol, el cual tiene tantas ramificaciones como argumentos. Cada ramificación se va conectando a nodos de niveles superiores. Los nodos hojas son siempre nodos terminales.

Existen varios ejemplos de conjunto de dominios tales como las operaciones aritméticas (+, -, *, /), funciones trigonométricas y logarítmicas (seno, coseno, tangente, logaritmo, exponencial), operaciones booleanas (*and*, *or* y *not*), operadores condicionales (*si*, *mientras*, *etc.*), funciones iterativas, funciones recursivas y cualquier función creada para un dominio específico que pueda definirse. El conjunto de los terminales está formado por las variables atómicas las cuales pueden representar entradas, valores constantes, sensores, detectores de estado o variables

Las expresiones son individuos de la población que deben cumplir con dos propiedades: la propiedad de cerradura (closure) y la propiedad de suficiencia.

a) Cerradura:

Significa que cualquier elemento de una expresión debe pertenecer al conjunto de funciones o al conjunto de terminales. Además, los argumentos con los que se llama una función deben de ser exactamente del mismo tipo que los parámetros definidos en dicha función

b) Suficiencia:

Significa que la solución del problema que buscamos debe poder ser formada por una combinación de funciones y terminales. Si no es posible representar la solución con los conjuntos de funciones y terminales definidos no podemos llegar a una solución

3.3.2.1 Pasos de la programación genética

La creación de árboles comienza con la elección de una función en forma aleatoria como nodo raíz, los siguientes elementos se seleccionan de una combinación de funciones y elementos terminales. El crecimiento de la rama de un árbol se termina cuando se selecciona un elemento terminal. Existen tres métodos para la creación de árboles, descritos por Koza (1992). Dichos métodos deben considerar las propiedades de cerradura y suficiencia. La raíz de los árboles siempre es un nodo función, así como todos los nodos internos, mientras que los nodos hoja siempre son elementos terminales. Los métodos para la generación de individuos son:

- a) Completo: Crea árboles con la altura máxima permitida en todas las hojas.
- b) Crecimiento: La altura de los árboles tiene un máximo el cual no puede sobrepasar, pero las alturas menores a ésta se obtienen dependiendo cuando se seleccione el elemento terminal por un proceso aleatorio

- c) **Mítad y Mítad:** Es una mezcla de los dos anteriores, donde la mitad se crea con el método completo y la otra mitad por el método de crecimiento

La función objetivo es la medida del desempeño de un individuo como solución potencial en el dominio del problema, generalmente resultado de la evaluación de un error, como el error cuadrático medio.

La *adaptabilidad* es la capacidad de sobrevivir y reproducirse. En la programación genética representa la fuerza de conducción de la población para llegar a sus objetivos. La aptitud o adaptabilidad se expresa al asignar un valor escalar a cada elemento de la población, que indica que tan bueno es el individuo. La aptitud puede asignarse al igual que en el algoritmo genético por los métodos de ordenación o escalamiento cuyos objetivos son poner la aptitud en valores mayores o iguales a 0, además de incrementar la diferenciación de aptitud muy cercanos.

Para algunos problemas complejos se requiere un alto uso de recursos computacionales que incluso puede sobrepasar la capacidad de los recursos de la máquina donde se ejecuta. Para la protección de recursos, la programación genética limita la altura de las estructuras que puede crear, así como su incremento como resultado de la aplicación de operadores genéticos.

La recombinación es un operador genético primario, cuyo objetivo es incrementar la variación creando nuevos individuos tomando partes de diferentes individuos. La recombinación puede modificar la altura máxima del árbol por lo cual se requiere de la definición de controles que limiten su crecimiento. Otro aspecto que hay que cuidar es la producción de individuos que a pesar de ser estructuralmente válidos no pertenecen al espacio de búsqueda; todas las expresiones deben satisfacer las propiedades de cerradura y de suficiencia. Existen cruza las cuales son poco atractivas como la recombinación donde intervienen las raíces de los padres, la recombinación entre terminales y la recombinación de individuos parecidos o incluso iguales. Sin embargo, la disminución en la diversidad es de mayor magnitud en los algoritmos genéticos. Esto trae como efecto principal la convergencia prematura o también conocida como supervivencia del mediocre que es en realidad una solución sub-óptima. El procedimiento de cruza implica dividir los individuos en dos fragmentos, en el fragmento de recombinación y el resto del árbol. La Figura 3.9 muestra un ejemplo de recombinación.

Existen otros operadores que se denominan secundarios los cuales no siempre se aplican. Por ejemplo, la mutación puede no aplicarse por que la recombinación, al seleccionar diferentes posiciones en cada progenitor, produce los efectos de la mutación que son mantener la diversidad y recorrer el espacio de búsqueda por estados no visitados. Cabe aclarar que para la mayoría de los problemas basta con la aplicación de los operadores primarios. Ejemplos de estos operadores

secundarios son: mutación el cual produce cambios aleatorios, siendo su principal objetivo incrementar la diversidad, si disminuye la diversidad provoca la convergencia prematura. La mutación es un operador asexual; el método de mutación más común es del tipo *grow* donde se selecciona un nodo en forma aleatoria, se hace la poda de toda la rama a partir del nodo seleccionado y se sustituye por otro sub-árbol generado al azar. Otros métodos de mutación son el *shrink* el cual reduce o compacta al sustituir nodos función por nodos terminales, el cíclico y la mutación de los terminales (Figura 3.10)

La programación genética en forma natural puede ejecutarse por siempre. Sin embargo, es necesario determinar criterios de finalización. Se pueden reconocer dos predicados uno generacional y uno de éxito. Tradicionalmente la terminación se basa en un máximo de ejecución de épocas.

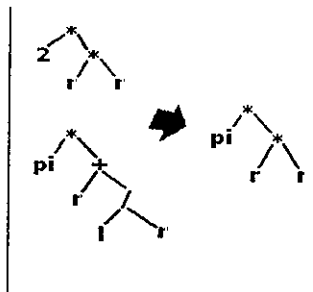


Figura 3.9: Recombinación

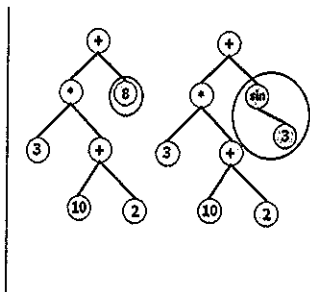


Figura 3.10: Mutación

3.3.2.2 Parámetros de control

La programación genética utiliza varios parámetros para ajustar el algoritmo para determinar su desempeño. Existen dos tipos de parámetros, los básicos y los secundarios. Los parámetros básicos son el tamaño de la población y el máximo número de generaciones. Entre los parámetros secundarios son la probabilidad de recombinación, probabilidad de mutación, la altura máxima de los árboles cuando es creado y la altura máxima cuando se aplican operadores genéticos. El i -ésimo árbol es representado por A_i , el conjunto de árboles representa la población A . $A(g) = (A_1, A_2, \dots, A_m)$, donde m es el tamaño de la población y g es el número de generación. La población inicial $A(0)$ es creada aleatoriamente, $F_i(g)$ es el valor de aptitud para el i -ésimo árbol en la g -ésima generación. Las siguientes generaciones son obtenidas por un porcentaje de selección y por la aplicación de los operadores genéticos de mutación y recombinación (Zitang, Ohm y Mühlenbein, 1997)

3.3.2.3 Aplicaciones de la programación genética

La programación genética se puede utilizar en problemas en los cuales la solución puede verse como la búsqueda de un programa dentro del espacio de posibilidades, el cual produzca una respuesta deseada a partir de ciertos estímulos. En muchas de las aplicaciones el problema se puede expresar en funciones matemáticas

Los campos y dominios que Koza (1992) expuso como ejemplos de la evidencia de la aplicación de la programación genética son en control óptimo, planeación, secuencia de la inducción, regresión simbólica, programación automática, generación de estrategias de juego-jugador, integración, diferenciación, problemas inversos, identidades matemáticas, entre otras.

3.4 Comparativo entre las técnicas expuestas

Para la programación genética se considera que sus mejores aciertos son:

- a) Encontrar relaciones funcionales descritas con ecuaciones
- b) La función objetivo puede considerar diferentes conocimientos o información acerca del fenómeno. Además, puede considerar un conjunto de funciones objetivo, conocida como multi-objetivos.
- c) En general, la programación genética puede ser un buen método de búsqueda para encontrar el mejor modelo dado un conjunto de funciones y terminales

Sus puntos más débiles:

- a) El punto débil es que sistemas que sufren cambios son más difíciles de adaptarse. Con la nueva información puede requerir iniciar de nuevo el proceso de identificación.

Los aciertos identificados para utilizar un GAFIS son:

- a) Con los resultados obtenidos pueden hacerse descripciones lingüísticas del fenómeno estudiado. Estas descripciones son expresadas en las reglas.
- b) Generan un menor número de divisiones comparado con la red ART.
- c) Cuando llegan nuevos datos no pierde totalmente el proceso de entrenamiento. Se puede mantener el número de variables, funciones de pertenencia, sus tipos y sus parámetros y solo adaptar los parámetros del consecuente.

Las desventajas identificadas son:

- a) Requiere de pre-procesos, como son el escalamiento y completar la información para realizar sus estimaciones.

- b) El número de parámetros a estimar en el consecuente suele ser grande o incrementarse exponencialmente cuando se añade una nueva variable de entrada o una función de pertenencia.
- c) Existe la posibilidad de encontrar puntos que no pueden ser estimados
- d) Es difícil de considerar conocimientos del fenómeno para hacer mejores estimaciones

Los aciertos de la red ART son:

- a) Sus tiempos de clasificación son rápidos, y no requieren un entrenamiento previo
- b) No pierde totalmente lo aprendido cuando llega un nuevo conjunto de datos. Solo modifica los pesos y argumentos donde los nuevos datos inciden

Sus desventajas:

- a) No es posible obtener información extra del fenómeno lo cual impide hacer una descripción de su dinámica o relaciones entre elementos.
- b) El número de patrones que se genera puede crecer indefinidamente
- c) Es difícil considerar conocimientos del fenómeno para mejores estimaciones

Basados en estas técnicas no convencionales y considerando sus ventajas y desventajas, se llevará a cabo la implementación de las mismas. Adicionalmente, se introducirá una representación alternativa de la programación genética la cual tiene como uno de sus objetivos generar estructuras matemáticas simples (principal problema de la programación genética) y eficientes. Para tal efecto, el siguiente capítulo describe a detalle dichas implementaciones para el problema específico de modelado de variables climáticas

Capítulo 4

4 Descripción de técnicas

En este capítulo se describen las técnicas no convencionales implementadas en este trabajo. Tres técnicas no convencionales fueron consideradas. La primera de ellas, la programación genética, es aplicada a un problema de regresión simbólica proponiéndose una modificación en la representación de los individuos. La finalidad de esta propuesta es facilitar la representación de polinomios, que expresen modelos que utilizan combinaciones lineales o no lineales. La segunda técnica es un sistema experto borroso basado en el modelo de Takagi-Sugeno, cuyo antecedente se determina por medio de algoritmos genéticos. A este tipo de método híbrido se conoce como GAFIS. La tercera técnica aplicada es la red ART2 con dos modificaciones; la primera de ellas define un peso diferente a cada entrada y la segunda modificación proporciona una capacidad de predicción. Las tres técnicas se aplicaron para predecir el comportamiento de variables meteorológicas, en específico la temperatura ambiental. Los datos fueron obtenidos de una estación meteorológica ubicada en el lago de Texcoco, en el Estado de México.

4.1 Método tradicional

4.1.1 Introducción

Existen diferentes métodos de predicción tradicional, uno de los más simples y utilizados es el método de la descomposición el cual considera a la serie de tiempo como resultado de la composición de varios factores que son la tendencia, la estación, los ciclos y las variaciones irregulares.

4.1.2 Descomposición

El método de la descomposición determina los diferentes componentes para determinar los valores futuros de la serie (y). Los valores futuros se calculan por la ecuación [4.1] donde T es la tendencia, C representa el componente cíclico, S el componente estacionario y I las irregularidades. La tendencia será calculada por mínimos cuadrados de los movimientos medios centrados. El método

de movimientos medios centrados se describe a continuación. Dada una serie x_1, x_2, \dots, x_n , se desea predecir m observaciones. El cálculo comienza con la suma de m movimientos (M_i) que se muestra en la expresión [4.2], después se calcula el movimiento centrado C_k (expresión [4.3]) y por último se calcula el movimiento centrado promedio (P_v) (expresión [4.4]).

$$y = T * C * S * I \quad [4.1]$$

$$M_i = \sum_{j=0}^{m-1} x_{i+j} \quad \text{para } i = 1, \dots, n-m+1 \quad [4.2]$$

$$C_k = M_k + M_{k+1} \quad \text{para } k = 1, \dots, n-m \quad [4.3]$$

$$P_v = \frac{C_{v-\frac{m}{2}}}{2 * m}, \quad \text{para } v = \left(\frac{m}{2}\right) + 1, \dots, n - \left(\frac{m}{2}\right) \quad [4.4]$$

El cálculo de la tendencia utiliza el movimiento centrado promedio. $P_v = f(s_v)$, siendo $s_v = \text{modulo}(v.m)+1$. La función generalmente es una línea, aunque se puede describir con otras funciones. Para funciones polinomiales esta función se observa en la ecuación [4.5]. Inicialmente, los coeficientes a_0, a_1, \dots, a_p son desconocidos y pueden ser determinados por regresión.

$$P_v = a_0 + a_1 s_v + \dots + a_p s_v^p \quad [4.5]$$

Una vez determinados los coeficientes se calculan los valores de T_w al aplicar la ecuación [4.5], siendo $w = 1, \dots, m$.

El componente estacionario se calcula en base a las ecuaciones [4.6] y [4.7], donde $u = n/m$. El componente estacionario (S_{qv}) se calcula con la ecuación [4.6], siendo w la posición del elemento en el ciclo [$w = \text{modulo}(i.m)$], q el número de repeticiones o ciclo [$q = (v-w)/m$] y el componente estacionario promedio (S_w) se calcula con la ecuación [4.7].

$$S_{qv} = \frac{x_v * 100}{P_v} \quad [4.6]$$

$$S_w = \frac{\sum_{j=1}^u S_{jw}}{|S_{jw}|} \quad [4.7]$$

Los componentes C e I son calculados mediante la ecuación [4.8]. Con la definición de todos los componentes conocidos se lleva a cabo la predicción de los valores futuros

$$C * I = \frac{x}{T * S} \quad [4.8]$$

4.2 Programación genética

La primera de las técnicas analizadas es la programación genética con una representación de polinomios la cual se detalla a continuación, así como los operadores genéticos asociados

4.2.1 Representación

La estructura para representar los modelos está formada por cuatro partes: los coeficientes, las ramas, una función de adición la cual suma todas las combinaciones propuestas, y la salida. Los coeficientes (parámetros de cada rama y el término constante) pertenecen a los números reales, existiendo un coeficiente por cada rama más uno. Las ramas son expresiones matemáticas representadas por medio de árboles, las cuales son combinaciones de los elementos del conjunto de primitivas descritos en la Tabla 4.1. La Figura 4.1 muestra la estructura propuesta en este trabajo. Dicha estructura permite representar modelos que utilizan combinaciones lineales o no lineales de términos lo cual permite representar modelos auto-regresivos (AR), promedio móvil (MA), lineales, no lineales (N), con presencia de ruido, con uso de variables exógenas (X) y sus distintas combinaciones. El modelo propuesto a utilizar es el NARX (no lineal, auto-regresivo con variables exógenas) que se muestra en la expresión [4 9], donde T es la temperatura, H es la humedad, R es la radiación solar, V es la velocidad del viento, D es la dirección del viento y h es la hora del día y \wedge indica valor estimado

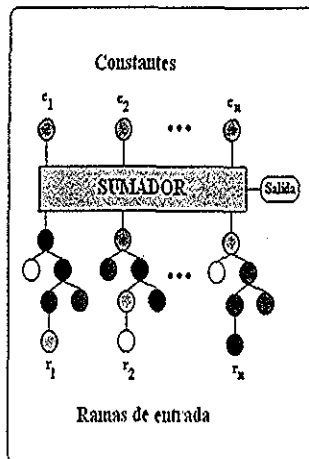


Figura 4.1: Estructura propuesta.

TESIS CON
FALLA DE ORIGEN

Dada la estructura, individuos de la población sólo pueden representar expresiones de la forma presentada en la ecuación [4.9]. Otros trabajos proponen el uso de gramáticas libres de contexto para limitar las expresiones que se pueden generar con la programación genética (Whigham y Crapper, 1999; Schaefer, 1999). Dado que la posición de los coeficientes en el arreglo es conocida pueden manipularse para calcularse de distintas formas. La programación genética tradicionalmente aproxima los valores de los coeficientes al aplicar los operadores genéticos, pero también se han aplicado métodos como el de retro-propagación, el *hill climbing* (ascenso de colinas) y el método de mínimos cuadrados (Whigham y Crapper 1999; Yoshihara *et al*, 2000; Lee, 1999; Lee, 2001; Rodríguez, 2001). La determinación de coeficientes para este trabajo se realizó por medio del método de mínimos cuadrados.

$$T_t = f(\hat{T}_{t-1}, \dots, \hat{T}_{t-n_T}, H_t, \dots, H_{t-n_H}, R_t, \dots, R_{t-n_R}, V_t, \dots, V_{t-n_V}, D_t, \dots, D_{t-n_D}, h_t, \dots, h_{t-n_h}) \quad [4.9]$$

El método de creación de los individuos utiliza el método de crecimiento descrito por Koza (1992) para crear las ramas, por lo que se requiere definir los conjuntos de las funciones que se asignarán a los nodos internos y el conjunto de elementos terminales que será posible asignar a los nodos hojas. El crecimiento de un árbol es acotado a una profundidad máxima. Para el conjunto de funciones es necesario definir el número de parámetros de cada función. Los parámetros que generalmente se definen en el algoritmo de programación genética son el tamaño de la población, número de épocas máximo, porcentajes de individuos que son generados por operadores genéticos, el porcentaje de mutación y el porcentaje de recombinación. La altura de los árboles al aplicar los operadores genéticos puede sufrir cambios. En esta implementación, los parámetros para control del tamaño de la expresión no son los tradicionalmente utilizados, ya que el control se hace en base a los parámetros de números de ramas y la altura máxima de la rama. El número máximo de ramas que puede tener cada individuo es un parámetro específico para esta estructura. Los valores de los parámetros utilizados para este trabajo se aprecian en la Tabla 4.1.

Tabla 4.1: Parámetros para el caso de la PG.

Conjunto	Valor
Funciones	* sin cos exp
Parámetros	2 1 1 1
Terminales:	$T_{i,1}$ $T_{i,2}$ $T_{i,3}$
T: temperatura.	$H_{i,1}$ $H_{i,2}$ $H_{i,3}$
H: humedad.	$R_{i,1}$ $R_{i,2}$ $R_{i,3}$
R: radiación.	$V_{i,1}$ $V_{i,2}$ $V_{i,3}$
V: veloc viento	$D_{i,1}$ $D_{i,2}$ $D_{i,3}$ h_i
D: direcc. viento	
h: hora del día	
t: tiempo	
parámetros	Valor
Tamaño de población	50
No. de épocas	200
No. máximo de ramas	5
Altura máxima de la rama	2
% de cruza	95
% de mutación	5
% nuevos individuos	98

La estructura también puede representarse como un árbol en el cual el nodo raíz es siempre la función que suma todas las ramas multiplicadas por sus coeficientes y la suma del coeficiente independiente. La aridad (número de argumentos) de esta función es calculada por medio de la expresión [4.10], donde A es la aridad y N el máximo número de ramas.

$$A = N * 2 + 1 \quad [4.10]$$

Cuando se crea un individuo, la raíz siempre es la función sumadora. A este nodo se le asignan A vértices; los primeros $N+1$ vértices corresponden a nodos terminales a los que se les asigna un número real. Los vértices restantes corresponden a las ramas. Las ramas son creadas por medio del método de crecimiento donde el tipo de cada nodo de la rama es asignado en forma aleatoria a partir de los elementos del conjunto de funciones y del conjunto de terminales. Si un nodo ha llegado a la altura máxima, se seleccionará un elemento exclusivo del conjunto de terminales. La Figura 4.2 muestra el proceso de creación de un individuo.

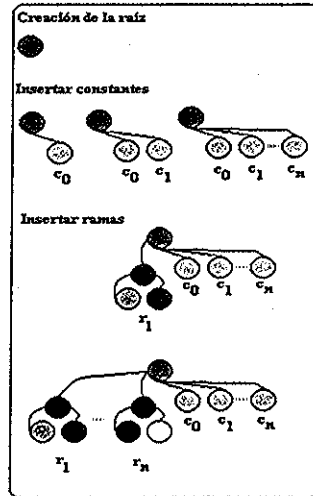


Figura 4.2: Proceso de Creación de un Individuo.

4.2.2 Operadores genéticos

Los operadores genéticos aplicados son la recombinación y la mutación. El proceso de recombinación presenta algunas modificaciones en comparación con el método tradicional el cual permite el punto de corte en cualquier parte del árbol. Para el caso de la propuesta PG de esta tesis, solo se permiten puntos de corte donde se unen las ramas con la función sumadora (intercambio de ramas entre individuos). Los padres son seleccionados al azar, se genera un número al azar x en el rango de $[0,1]$. Si x es mayor que el porcentaje de recombinación los cromosomas pasan intactos a la siguiente generación; en caso contrario, se aplica el proceso en el cual se selecciona aleatoriamente el punto de corte para cada padre, se poda las ramas y se intercambian, como se muestra en la Figura 4.3. El proceso de mutación aplicado es una modificación al proceso tradicional, donde cada individuo se somete a la probabilidad de ser mutado. Si se determina que no se mutará, pasa directamente a la siguiente generación; en caso contrario, se selecciona el punto de corte (solo en nodo función que este en contacto con el nodo raíz), el subárbol a partir del nodo seleccionado es podado y desechado. En la posición vacante se inserta otra nueva rama generada al azar. En la Figura 4.4 se muestra el proceso de mutación.

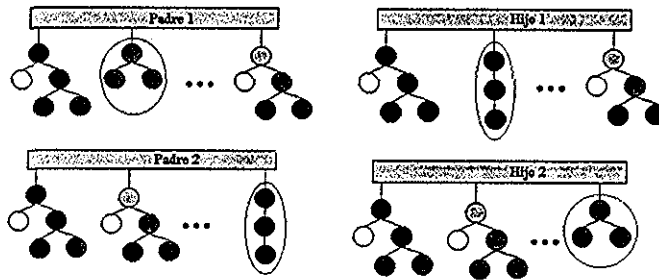


Figura 4.3: Operación de recombinación.

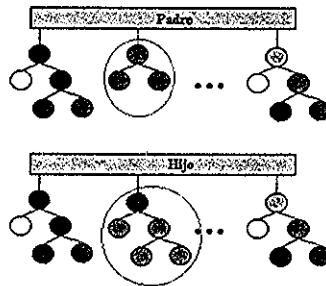


Figura 4.4: Operación de mutación.

4.2.3 Evaluación de los individuos

Cada una de las ramas son sub-expresiones matemáticas que son evaluadas como tal, el resultado de la evaluación r_{ij} de la i -ésima rama y la j -ésima observación para $i=1, \dots, s$ y $j=1, \dots, m$, siendo s el número de ramas y m el número de observaciones, r está en función de los n elementos no terminales (x_j) como lo muestra la ecuación [4.12]. El valor estimado de la variable dependiente y_j se calcula como la suma de los productos entre los coeficientes c_i por r_{ij} , como se muestra en la expresión [4.13]. El arreglo matricial se muestra en la Figura 4.5. Los coeficientes se determinan por mínimos cuadrados; una vez conocidos los coeficientes se estiman los valores de la variable dependiente como se muestra en la expresión [4.15]. El valor v que regresa es la norma infinito entre los valores observados y los valores estimados como lo muestra la expresión [4.16].

TESIS CON
 FALLA DE ORIGEN

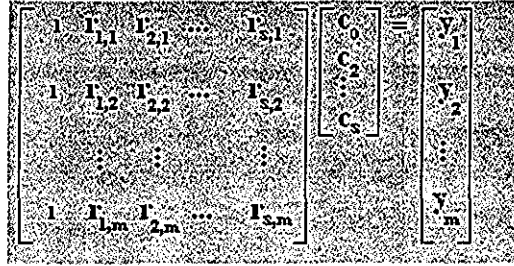


Figura 4.5: Representación matricial de los árboles.

$$R * C = Y \quad [4.11]$$

$$r_{ij} = f([x_1, \dots, x_n]) \quad [4.12] \quad C = \text{inv}(R'R)(R'Y) \quad [4.14]$$

$$y_j = c_0 + \sum_{i=1}^s c_i * r_{ij} \quad [4.13] \quad \hat{Y} = R * C \quad [4.15]$$

$$v = \max(|Y - \hat{Y}|) \quad [4.16]$$

4.3 Teoría de resonancia adaptativa

La red ART2 propuesta tiene dos modificaciones en relación al diseño propuesto por Carpenter *et al* (1988). La primera de ellas es considerar que no todas las variables de entrada tienen igual importancia para predecir la variable de salida, por lo cual proponemos un peso para ajustar las diferencias en importancia. Cuando se añaden parámetros a una técnica puede mejorar el desempeño, pero también plantea el problema de cómo determinar dichos pesos. Por tanto, la segunda consideración es ajustar todos los parámetros de la red por medio de los algoritmos genéticos. La aplicación de la red ART 2 es la clasificación no supervisada, por lo cual añadimos una propiedad de predicción. Existen redes ARIMAP, con funciones específicas de predicción, pero utilizan dos redes ART lo cual incrementa los requerimientos computacionales.

4.3.1 Algoritmo genético: Determinación de parámetros

Los individuos de la población representan los parámetros que requiere la red ART2 para operar; el primer gen representa el parámetro de vigilancia cuando la red ajusta pesos, el segundo parámetro es el parámetro de vigilancia cuando utilizamos a la red para estimar los valores de salida. La red

ART2 tradicional no utiliza entrenamiento por lo que en el momento de clasificar un vector de entrada también ajusta los pesos de la red. Para la red ART2 se consideran dos procesos El primero de ellos se presenta cuando se introducen nuevos datos y se ajustan el número de patrones y sus pesos El otro proceso considerado es cuando utilizamos la red para predicción, donde no se afecta la estructura de la red. Como se consideran dos procesos, también consideramos dos parámetros de vigilancia. En la propuesta tradicional no existe la diferenciación del proceso; cada nuevo vector de entrada primero se evalúa y después ajusta su estructura por lo que solo utiliza un solo parámetro de vigilancia La razón de este cambio es debido a que la entrada para esta aplicación es una ventana de tiempo la cual está constituida por varios vectores de entrada. Los genes restantes corresponden a cada variable de entrada y representan el peso de acuerdo a la importancia que tiene cada variable para predecir la salida La suma de los pesos es igual a uno y todos los pesos son positivos mayores o iguales a 0

Siendo una aplicación de determinación de parámetros, los parámetros del algoritmo y operadores usados son los que se proponen para el algoritmo genético. En la Tabla 4.2 se muestran los parámetros y sus valores usados para realizar las corridas. Los operadores genéticos aplicados son la recombinación lineal y la mutación para poblaciones con valores reales (Mühlenbein y Schlierkamp-Voosen, 1993).

Tabla 4.2: Parámetros para el caso de la red ART 2.

Conjuntos	Valores
<i>Variables independientes*</i>	<i>temperatura, humedad, radiación solar, vel del viento, dir. del viento</i>
<i>Variables dependientes</i>	<i>temperatura</i>
<i>parámetros</i>	<i>Valores</i>
<i>Tamaño de la población</i>	50
<i>No. de épocas</i>	200
<i>% de cruzamiento</i>	95
<i>% de mutación</i>	5
<i>% de nuevos individuos</i>	98

*valores anteriores

4.3.2 Evaluación de los individuos

Los datos se dividen en dos partes La primera mitad de los datos se utiliza para fijar los pre-pesos de la red ART; los datos restantes se utilizan para fijar el parámetro de vigilancia durante el proceso de estimación de la variable dependiente Los criterios considerados para evaluar cada propuesta

fueron el error absoluto máximo entre la estimación y el valor real, y el número de elementos que quedaron sin clasificar, aplicados tanto para el proceso de determinación de los pre-pesos como para la estimación. El valor de retorno se obtuvo considerando las 4 funciones objetivo (dos criterios por cada uno de los conjuntos en los que se dividieron los datos) por medio de una técnica de asignación de aptitud basada en el ordenamiento por dominancia. Un ejemplo de asignación por dominancia se observa en la Figura 4.6

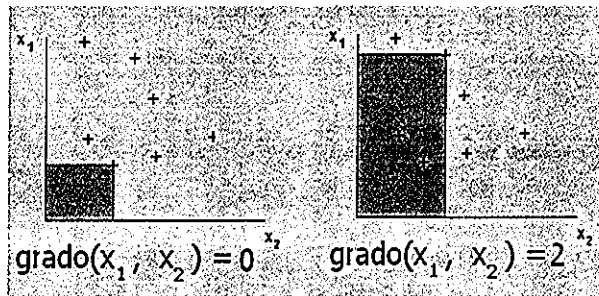


Figura 4.6 Ejemplo de asignación por dominancia.

4.3.3 Operación

En esta sección se describe el funcionamiento de la red ART2. El proceso inicia con la determinación del parámetro de vigilancia. Dado que las variables de entrada pueden tener diferente peso para la predicción de la temperatura, cada variable se pondera por un peso, el cual llamaremos pre-peso. El parámetro de vigilancia en las redes ART2 tradicionalmente se calcula por un proceso de prueba-error, mientras que los pre-pesos no han sido considerados. La alternativa propuesta en este trabajo es la determinación del parámetro de vigilancia y los pre-pesos por medio de algoritmos genéticos. El cromosoma propuesto se observa en la Figura 4.7, el cual está formado por G genes, dos para los parámetros de vigilancia y los restantes para cada pre-peso. Una representación real es usada en la codificación de los pre-pesos.

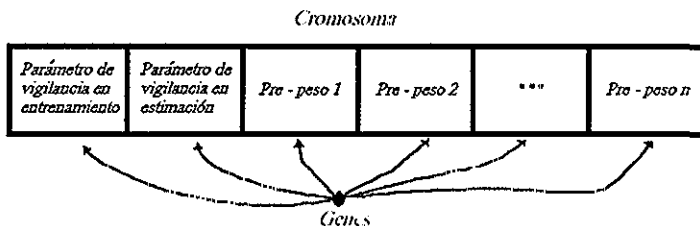


Figura 4.7: Cromosoma utilizado en el algoritmo genético.

TESIS CON
 FALLA DE ORIGEN

Una vez determinados los parámetros de vigilancia y los pre-pesos se entrena la red. La red propuesta es una red ART2 con una modificación que le permita predecir. La estructura de esta red se detalla en la Figura 4.8. Los componentes extra a la red ART2 son una entrada que contiene la variable a predecir (I), la salida estimada (O), las conexiones de las neuronas S_{sj} con la salida estimada y las conexiones de la variable a predecir con las neuronas S_{sj} . El entrenamiento es semejante al descrito en el capítulo anterior, con la diferencia que después de ajustar los pesos w_{ij} , se multiplican las salidas de S_{sj} por la entrada I , pero solo la neurona ganadora s tiene como salida 1; las neuronas restantes tienen un valor de salida 0. El ajuste de los pesos $X_{s(t+1)}$, en el tiempo $t+1$, se realiza por medio de la ecuación [4.17] donde m_s es el número de patrones que tiene el *cluster* (grupo)

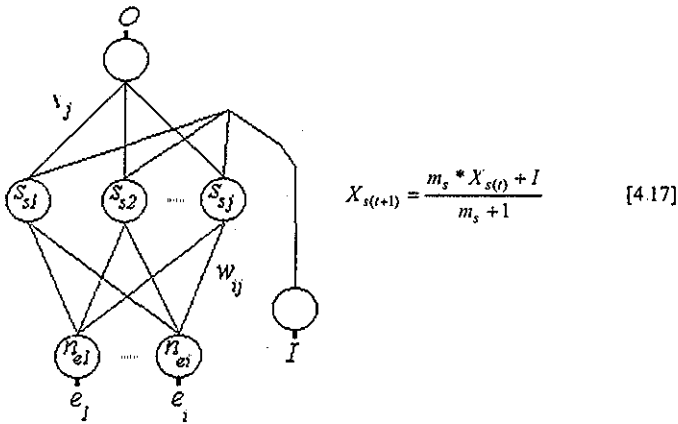


Figura 4.8: Estructura de la red propuesta

La Figura 4.9 muestra lo ocurrido durante el proceso de entrenamiento cuando un patrón no pertenece a ningún grupo de patrones. Al igual que la red ART2 original se genera un nuevo grupo, pero además se generan las conexiones extras. Las conexiones que se crean se marcan en la Figura 4.9 con línea punteada. Después del entrenamiento, la red se utiliza para predecir. En la Figura 4.10 se muestra la estructura de la red para predecir, durante esta fase no hay ajuste de pesos. Cuando predice multiplica las salidas de las neuronas (S_{sj}) por los pesos X_j sumándose todos los resultados en la neurona O aunque en términos prácticos solo se envía a la salida de la neurona O el peso X_g , siendo g la neurona ganadora

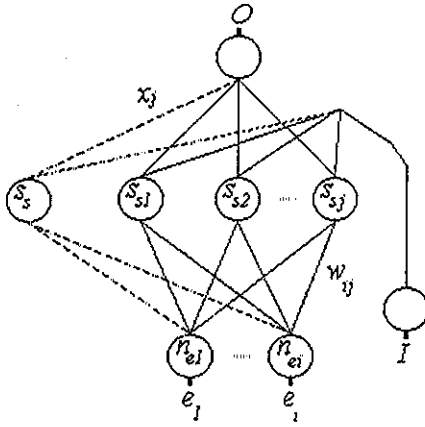


Figura 4.9: Generación de un nuevo grupo de patrones.

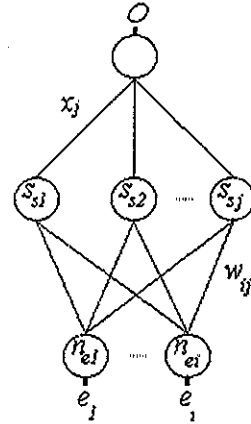


Figura 4.10: Estructura de la red para predicción

4.4 Sistema borroso de inferencia apoyado con algoritmos genéticos (GAFIS)

La aplicación del algoritmo genético tiene como finalidad obtener el antecedente de las reglas del sistema borroso. Con el modelo de Takagi-Sugeno (1985), el antecedente es la única parte del sistema que es obtenida por prueba y error, las otras partes del sistema tienen una forma precisa de determinarse. Si bien existen otras implementaciones de sistemas borrosos de inferencia evolutivos, varían en la estructura de cromosomas dobles presentada en este trabajo.

4.4.1 Algoritmo genético: Determinación del antecedente

Los individuos están formados por dos vectores (cromosomas), el vector de los parámetros de las funciones de pertenencia y el vector de los tipos de función. En el cromosoma de los parámetros, los genes están formados por tripletas las cuales representan los parámetros de una función de pertenencia triangular. El número de genes de este cromosoma está dado por el número de variables de entrada multiplicado por el número máximo de funciones de pertenencia por variable. En el cromosoma de los tipos, cada gen corresponde a un elemento del vector. Los alelos de este cromosoma son múltiples, representados por un número entero en el rango $[0, N_o \text{ de tipos de funciones}]$, un número para cada tipo y 0 cuando la función de pertenencia no aplica. Ambos

cromosomas poseen la misma longitud, por lo que cada gen en el cromosoma de parámetros está relacionado con uno y solo un gen del cromosoma de tipos de función. La codificación es un número entero para el tipo de función de pertenencia (tmf_j) y los parámetros (pm_k) de la función de pertenencia tienen una representación real. El individuo está formado por dos cromosomas, uno para representar las funciones de pertenencia y otro para representar sus parámetros. Esta representación se observa en la Figura 4.11, donde Var_i son las variables de entrada, mf_j corresponden a las funciones de pertenencia, siendo i la i -ésima variable (para $i=1, \dots, m$) y j la j -ésima función de pertenencia (para $j=1, \dots, n$) y k el k -ésimo parámetro de la función de pertenencia para $k=1,2,3$

Individuo

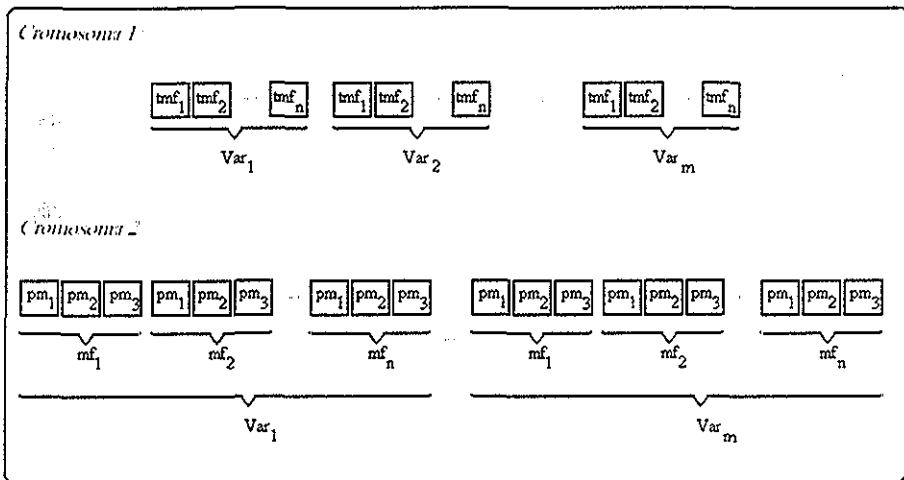


Figura 4.11: Representación del antecedente de GAFIS, utilizada por el algoritmo genético.

Para el sistema borroso se tienen como entradas las diferentes variables del estado del tiempo consideradas, como salida se tienen la temperatura esperada y el conjunto de posibles funciones de pertenencia. En la Tabla 4.3 se muestran los parámetros y los valores de dichos parámetros.

Los individuos son creados en forma aleatoria. Las funciones de pertenencia se distribuyen en todo el rango de la variable, dando un espacio igual para cada una de las funciones de pertenencia. El valor máximo de cada función de pertenencia es 1, y se encuentra ubicado en el espacio asignado a esa función de pertenencia. Los límites de sub-rango están expresados en la ecuación [4.18], donde nD es el número de división, nF es el número máximo de funciones de pertenencia, Ls es el límite superior, Li es el límite inferior, Mx es el valor máximo y Mn el valor mínimo del rango de la variable. La posición del parámetro central nT se obtiene con la expresión [4.19], donde nA es un

número al azar en el rango [0.01, 0.99] El parámetro izquierdo pI se obtiene con la expresión [4.20] y el parámetro derecho pD con la expresión [4.21], donde C es una constante representando la distancia máxima entre el parámetro central y los parámetros de los extremos. En el cromosoma de tipo de función de pertenencia, a cada gen se le asigna en forma aleatoria un número entero en el rango [0, nF], siendo nF el número de tipos de función de pertenencia. Existe una tabla la cual asigna un número entero a cada función de pertenencia.

Tabla 4.3: Parámetros y valores para el método GAFIS.

Conjuntos	Valores
Funciones de pertenencia	triangular, trapezoidal, gaussianas, gaussiana doble, sigmoides, sigmoides doble, prod Sigmoides, curva campana, curvas Z,S,P
Entradas	temperatura anterior, humedad anterior y la hora del día
Variable dependiente	temperatura
parámetros	Valores
Tamaño de la población	50
Máximo número de épocas	50
% Nuevos individuos	98
Máximo No. de funciones de pertenencia	3
% Cruza	95
% mutación	5

$$\{L_i, L_s\} = \left\{ Mn + \frac{(Mx - Mn)(nD - 1)}{nF}, Mn + \frac{(Mx - Mn) * nD}{nF} \right\} \quad [4.18]$$

$$nT = Li + (Ls - Li) * nA \quad [4.19]$$

$$pI = nT - nA * C \quad [4.20]$$

$$pD = nT + nA * C \quad [4.21]$$

Se forman todas las reglas completas utilizando solo el operador *AND*, definiendo las reglas como todas las posibles combinaciones que se pueden hacer utilizando todas las variables de entrada, con sus diferentes funciones de pertenencia. El número total de reglas nR que podemos tener se calcula con la expresión [4.22], donde nV es el número de variables:

$$nR = nF^{nV} \quad [4.22]$$

4.4.2 Operación

A continuación se detalla la manera de como GAFIS opera. Considerar x_{ij} como la i -ésima observación de la j -ésima variable independiente, y_i como la i -ésima observación de la variable dependiente para $i = 1, \dots, p$ y $j = 1, \dots, n$. X es la matriz de las variables (ver Figura 4.12) z_{jkl} son los parámetros de la k -ésima función de pertenencia del l -ésimo individuo para la j -ésima variable independiente, la cual se calcula con expresión [4 23] para $k = 1, \dots, q$ y $l = 1, \dots, r$. s_{ljk} es el valor de la k -ésima función de pertenencia, para el l -ésimo individuo, de la j -ésima variable e i -ésima observación. m es la función de pertenencia. El antecedente de las reglas se calcula con la combinación de los valores de una función de pertenencia por cada una de las variables independientes teniendo la forma de la expresión [4.24], donde $V_{l,m}$ es el valor convertido a borroso del antecedente de la m -ésima regla de la i -ésima observación, g es una función implicación y m es el número de reglas. El consecuente (C) es un polinomio de grado t , donde $t \in \{Z^+, 0\}$. Para fines de este trabajo solo se consideró $t = 1$, por lo que el consecuente tiene la forma dada por [4 25], donde a son los coeficientes del polinomio. El valor no borroso se obtiene por medio de la expresión [4 26].

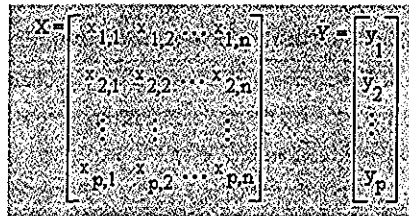


Figura 4.12: Matriz de entrada.

$$S_{ljk} = \mu(x_{ij}, z_{jkl}) \quad [4 23]$$

$$V_{l,m} = g(S_{1kk}, S_{2kk}, \dots, S_{lkk}) \quad [4 24]$$

$$C_{i,m} = a_{m,0} + a_{m,1} x_{i1} + \dots + a_{m,t} x_{it} \quad [4 25]$$

$$\hat{Y}_i = V_{i1} * C_{i1} + \dots + V_{ir} * C_{ir} \quad [4.26]$$



4.4.3 Cálculo de coeficientes

Durante el proceso de entrenamiento los coeficientes del consecuente son ajustados por medio del método de mínimos cuadrados. A son todos los términos conocidos de la expresión del lado derecho; esto es, los valores de V y los valores de las variables de entrada x . w está formada por un vector columna con todos los coeficientes a , que son desconocidos y B es un vector columna con todos los valores esperados de la salida. Esta relación se expresa en la expresión [4 27]

$$Aw = B \quad [4 27]$$

El resultado esperado en este caso es la obtención de la estructura de los sistemas borrosos, la cual se describe en el antecedente como el número de variables independientes, el número de funciones de pertenencia de cada variable independiente, los parámetros de las funciones de pertenencia y el tipo de la función de pertenencia. Un resultado temporal son las funciones del consecuente; con la entrada de nuevos datos históricos el consecuente puede reajustarse. Por último, esperamos obtener estimaciones de los valores reales presentes o futuros utilizando datos históricos de las variables independientes.

4.5 Casos de prueba

En esta sección se describirán las pruebas realizadas en este trabajo. El primer caso corresponde al método convencional de la descomposición. En los casos 1 a 3 se utilizó la técnica de programación genética con la representación polinomial, con la red AR12 predictiva se probaron los casos 4, 5 y 6. Por último, el GAFIS fue aplicado para el caso 7.

4.5.1 Tradicional

El método tradicional aplicado para el pronóstico de la temperatura ambiente es el conocido como el *método de la descomposición*. La serie de tiempo consideró 5 días, lo cual corresponde aproximadamente al 15% de las observaciones. Con la información de los 5 días se trata de estimar el siguiente día. Se realizaron dos corridas, donde la ventana de tiempo solo utiliza los valores medidos para realizar los pronósticos y la otra donde los valores estimados son considerados en la ventana de tiempo para realizar el pronóstico. El ciclo está compuesto de 96 datos correspondientes a las observaciones de un día, considerándose la media móvil.

4.5.2 Programación genética

Para todos los casos con programación genética se utilizaron los datos sin escalar. La variable a predecir fue el incremento de temperatura. Con el incremento de temperatura se calculó la temperatura en $t+1$. Se consideraron tres grupos de datos. El 15% de los datos se utilizaron para generar los modelos. Posteriormente, se realizaron dos pruebas: (1) con un grupo del 60% de los datos, conteniendo tanto datos que se utilizaron para determinar el modelo como datos nuevos, y (2) grupo con el 40% de los datos no utilizados. Para cada grupo se utilizaron los valores estimados para calcular los siguientes valores. Se calcularon los valores máximos, mínimos y promedios de cada error y su distribución. La descripción de cada uno de los casos de prueba para esta técnica se presenta a continuación:

- Caso 1: La característica de esta corrida fue la consideración de la evaluación de los individuos en función a la norma infinito entre los valores estimados y los valores reales.
- Caso 2: Por último, se realizó la evaluación de los individuos en función a la norma uno entre los valores estimados y los valores reales.
- Caso 3: Para este caso, la evaluación de los individuos se realizó en función a la norma uno y la norma infinito, simultáneamente, entre los valores estimados y los valores reales.

4.5.3 Red ARI 2

Con el 15% de los datos se obtuvieron los pesos y los parámetros de vigilancia por medio de algoritmos genéticos. Con el 60% de los datos, incluyendo el 15% antes mencionado, se ajustaron los pesos de la red y con el 40% de los datos no utilizados, se probó el desempeño de la red. Todos las predicciones consideraron los datos estimados. Las variantes de prueba para esta técnica son descritas en los siguientes casos:

- Caso 4: Utilizando los datos no normalizados, se intentó predecir el comportamiento de la temperatura ambiente.
- Caso 5: Con datos escalados en un rango de $[0,1]$, se realizó la predicción del incremento de temperatura el cual se utilizó para la estimación de la siguiente temperatura.
- Caso 6: Con los datos reales se predijo el incremento en temperatura directamente.

4.5.4 Algoritmos genéticos y sistema de inferencia borroso

Con el 15% de los datos se obtuvieron los sistema borroso de inferencias por medio de algoritmos genéticos. Con el 60% de los datos, incluyendo el 15% antes mencionado, se ajustaron los parámetros de los consecuentes de las reglas y con el 40% de los datos restantes se probó el desempeño del sistema. La primera parte considera las estimaciones a corto plazo y posteriormente se realizan predicciones a largo plazo.

- Caso 7: La estimación se realizó con datos normalizados y prediciendo el incremento de temperatura. La información con la que se ajustaron los parámetros del consecuente recibió un pre-proceso el cual consistió en interpolar la información. El primer paso consistió en normalizar los datos en un rango de $[0,1]$, por cada variable de entrada. Cada variable se dividió en 25 secciones y se intentó estimar valores por interpolación para las diferentes combinaciones entre las 3 variables. Se eliminaron todos los vectores que no pudieron ser interpolados y los sobrantes fueron considerados como entrada para la determinación del sistema de inferencia borroso.

Basados en los casos de prueba descritos para cada una de las técnicas, en el capítulo siguiente se presenta un análisis de los resultados obtenidos.

Capítulo 5

Los dos procesos de predicción realizados fueron PCP (predicción a corto plazo) y PLP (predicción a largo plazo). La PCP no incluye retroalimentación, y solo puede predecir la siguiente temperatura. En la PLP si hay retroalimentación, los valores estimados son considerados para realizar la siguiente estimación.

5 Resultados

5.1 Caso 0:

En la Figura 5.1 se presenta la distribución de los errores entre las estimaciones y los datos observados para PCP. La diferencia entre los datos reales y los estimados se ejemplifica en la Figura 5.3. Para PLP se muestra su distribución de los errores en la Figura 5.2 y un ejemplo en la Figura 5.4. Las normas para PCP son de 6765, 152.6 y 9.268 para la norma 1, norma 2 y norma infinito, respectivamente. Para la PLP se obtuvo una norma 1 de 6.278×10^4 , 1403 de norma 2 y 56.73 de norma infinito.

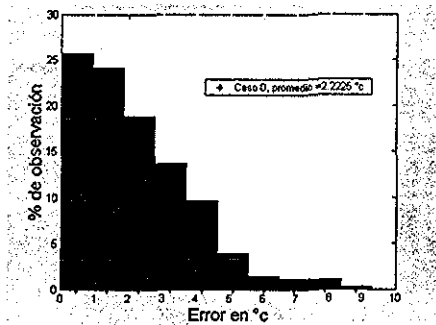


Figura 5.1: Distribución del error, para PCP.
No de observaciones: 3072

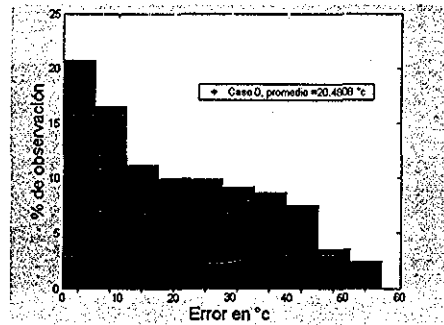


Figura 5.2: Distribución del error, para PLP.
No de observaciones: 3072

TESIS CON
FALLA DE ORIGEN

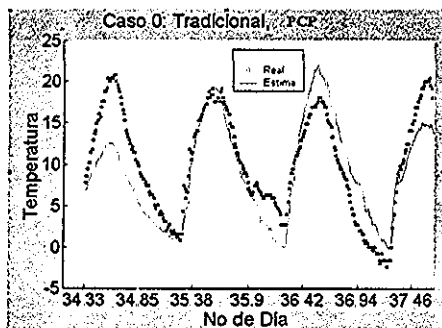


Figura 5.3: Comparación entre las estimaciones y las observaciones para PCP en °C.

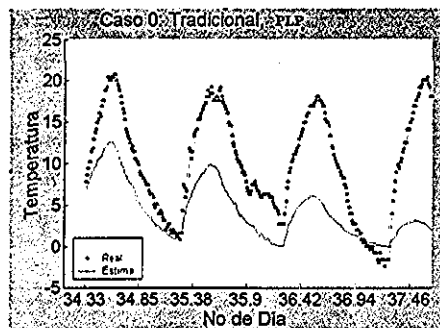


Figura 5.4: Comparación entre las estimaciones y las observaciones para PLP en °C.

5.2 Caso 1:

En el caso 1 correspondiente a la PG, la función objetivo se definió por medio de la norma infinito realizándose 20 corridas. El modelo con mejor desempeño se describe con la expresión [5.1] y su expresión equivalente [5.2]. Los modelos con mejor desempeño por corrida se presentan en el anexo A. En la Tabla 5.1 se muestra un resumen de los errores obtenidos entre las observaciones y las estimaciones para PCP, bajo diferentes normas, así como también el número de nodos. En el anexo B se presentan los resultados de cada una de las corridas con los que se generó la Tabla 5.1. Las Tablas 5.2 a la 5.4 presentan los diferentes tipos de errores, para los diferentes grupos de datos, para PLP. Los resultados por corrida, para PLP, se presentan en el anexo C. Una parte de las estimaciones hechas con el modelo [5.1], como ejemplo, se muestran en las Figuras 5.5, 5.6 y 5.7.

$$(\text{modelo } t_{-1} (-t_{-3} t_{-1}) t_{-3} (-t_{-1} t_{-3}) (\sin H) 15.7325 7.6939 -15.7414 -7.6439 0.10088 0.10258) \quad [5.1]$$

$$t_i = 0.10258 + 0.3947 t_{-1} - 0.4036 t_{-3} + 0.10088 \sin(H) \quad [5.2]$$

Tabla 5.1: Caso 1, Errores y nodos. PCP.

Corrida	Norma 1	Norma 2	Norma infinito	No nodos
mínimo	136.4907	10.1565	1.9826	13
media	148.9289	11.0517	2.083	15.65
máximo	162.7383	11.889	2.1454	19
desviación	6.1507	0.4784	0.0465	1.59852
c.v.	0.0413	0.0433	0.0223	0.1021

*c.v. coeficiente de variación

Tabla 5.2: Valores de Norma 1, PLP.

Corrida	grupo 1	grupo 2	grupo 3
mínimo	136.5	910.1	732.9
media	153.4	1020	895.3
máximo	170.5	1122	1118
desviación	6.44	47.4	88.64
c.v.	0.0420	0.0465	0.0990

Tabla 5.3: Valores de Norma 2, PLP.

Corrida	grupo 1	grupo 2	grupo 3
mínimo	10.28	25.9	25.75
media	11.42	29.05	31.01
máximo	12.61	31.43	38.77
desviación	0.4228	1.378	2.994
c.v.	0.0370	0.0474	0.0965

Tabla 5.4: Valores de Norma Infinito, PLP.

Corrida	grupo 1	grupo 2	grupo 3
mínimo	1.979	2.837	2.975
media	2.12	3.252	3.992
máximo	2.719	4.665	4.803
desviación	0.1578	0.5049	0.4931
c.v.	0.0744	0.1553	0.1235

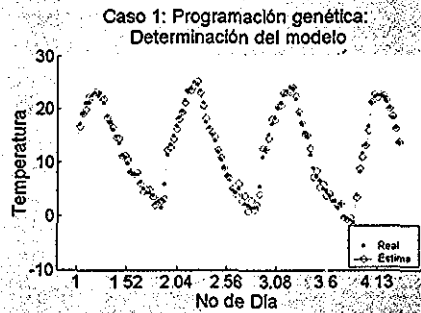


Figura 5.5: Estimaciones sobre las observaciones del grupo 1.

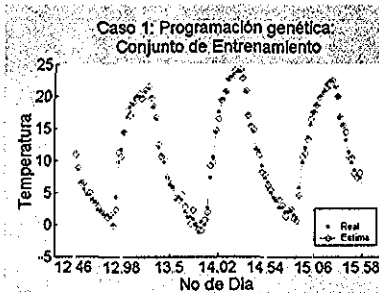


Figura 5.6: Estimación sobre las observaciones de entrenamiento (grupo 2).

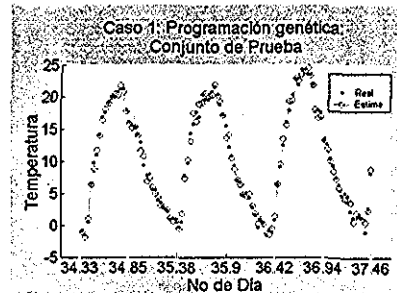


Figura 5.7: Estimación sobre las observaciones de prueba (grupo 3).

5.3 Caso 2:

En el caso 2 se aplicó programación genética teniendo como función objetivo la norma 1. El modelo que presentó el mejor desempeño de las corridas realizadas se describe con la expresión [5 3] y su equivalente en la expresión [5 4]. Para las corridas restantes, las expresiones que representan el mejor desempeño se muestran en el anexo D. En la Tabla 5 5 se muestran los errores obtenidos con los datos del grupo de entrenamiento para PCP. Las Tablas 5 6 a la 5 8 muestran los

diferentes tipos de errores para los diferentes grupos para PLP. Una parte de las estimaciones del modelo [5.3] se muestran en las Figuras 5.8, 5.9 y 5.10. En el anexo E se presentan los errores por corrida para PCP y para PLP se exhiben en el anexo F.

$$\begin{aligned}
 &(\text{modelo } (* r_{t-2} t_{t-2}) (* r_{t-1} t_{t-1}) r_{t-1} (+ t_{t-3} r_{t-1}) r_{t-2} 0.00087437 - 0.00092556 0.066203 - 0.038852 - \\
 &0.024251 0.045344) \quad [5.3] \\
 t_t = &0.045344 + 0.00087437 r_{t-2} t_{t-2} - 0.00092556 r_{t-1} t_{t-1} - 0.038852 t_{t-3} + 0.027351 r_{t-1} - 0.024251 r_{t-2} \\
 &[5.4]
 \end{aligned}$$

Tabla 5.5: Errores y nodos. PCP.

Corrida	Norma 1	Norma 2	Norma infinito	No nodos
mínimo	117.1992	8.8531	2.2276	12
media	133.2591	9.9352	2.3081	16.05
máximo	136.7738	10.1572	2.5269	20
desviación	5.5921	0.3791	0.0771	2.35025
c.v.	0.0420	0.0382	0.0334	0.1464

Tabla 5.6: Valores de Norma 1, PLP.

Corrida	grupo 1	grupo 2	grupo 3
mínimo	127.2	884	626.2
media	169.9	1231	997.6
máximo	299.6	3822	4289
desviación	45.25	722.6	910.5
c.v.	0.2663	0.5870	0.9127

Tabla 5.7: Valores de Norma 2, PLP.

Corrida	grupo 1	grupo 2	grupo 3
mínimo	9.758	25.69	22.24
media	13.5	49.2	45.31
máximo	31.84	292.2	287.1
desviación	5.903	63.33	64.84
c.v.	0.4373	1.2872	1.4310

Tabla 5.8: Valores de Norma infinito, PLP.

Corrida	grupo 1	grupo 2	grupo 3
Mínimo	2.242	2.766	2.74
Media	3.173	9.434	7.721
Máximo	8.781	79.41	64.11
desviación	1.703	17.66	14.81
c.v.	0.5367	1.8720	1.9181

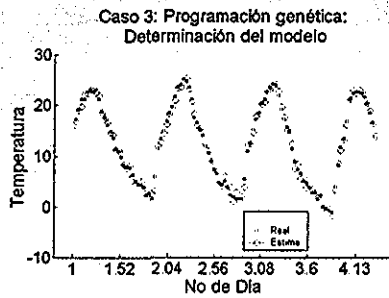


Figura 5.8: Estimación sobre las observaciones del grupo 1.

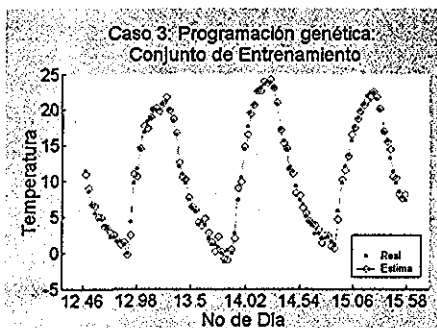


Figura 5.9: Estimación sobre las observaciones de entrenamiento (grupo 2).

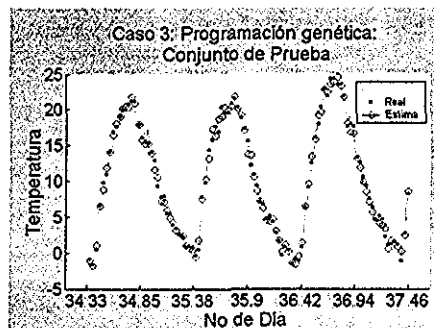


Figura 5.10: Estimación sobre las observaciones de prueba (grupo 3).

5.4 Caso 3:

En el caso 3 se aplicó programación genética considerando tanto la norma 1 como la norma infinito como función objetivo. El mejor modelo de todas las corridas realizadas se describe con la expresión [5.5] y su equivalente [5.6]. Para las corridas restantes, las expresiones con mejor desempeño se muestran en el anexo G. En la Tabla 5.9 se muestran los errores para PCP. Las Tablas 5.10 a la 5.12 muestran los diferentes tipos de errores para PLP para los diferentes grupos. Las estimaciones del mejor modelo se muestran en las Figuras 5.11, 5.12 y 5.13. Los anexos H e I muestran los errores para PCP y para PLP, respectivamente.

$$(modelo (+ d_{t,2} h_{t,2}) (\sin r_{t,2}) (- h_{t,3} h_{t,2}) t_{t,3} (\cos r_{t,3}) 0.0038188 0.055095 0.10502 -0.01781 -0.26549 0.14237) [5.5]$$

$$t_t = 0.1437 + 0.0038188 (d_{t,2} h_{t,2}) + 0.05509 \sin(r_{t,2}) + 0.10502 (h_{t,3} h_{t,2}) - 0.01781 t_{t,3} - 0.26549 \cos(r_{t,3}) [5.6]$$

Tabla 5.9: Caso 3. Errores y nodos, PCP.

Corrida	Norma 1	Norma 2	Norma infinito	No nodos
mínimo	133.61	9.8212	2.0331	10
media	144.6331	10.7313	2.1146	15.7
máximo	154.0301	11.4878	2.1812	19
desviación	6.4623	0.5276	0.046	2.47301
c.v.	0.0447	0.0492	0.0218	0.1575

TESIS CON FALLA DE ORIGEN

Tabla 5.10: Valores de Norma 1, PCP.

Corrida	grupo 1	grupo 2	grupo 3
mínimo	137.9	908.2	687.1
media	150.8	985.1	885.5
máximo	158.3	1067	1109
desviación	5.551	38.99	138
c.v.	0.0368	0.0396	0.1558

Tabla 5.11: Valores de Norma 2 PCP.

Corrida	grupo 1	grupo 2	grupo 3
mínimo	10.33	26.04	23.98
media	11.22	28.25	30.61
máximo	11.69	30.84	38.42
desviación	0.4029	1.164	4.685
c.v.	0.0359	0.0412	0.1531

Tabla 5.12: Valores de Norma infinito, PCP.

Corrida	grupo 1	grupo 2	grupo 3
Mínimo	2.034	2.865	2.878
Media	2.209	3.282	3.807
Máximo	2.53	4.369	4.674
Desviación	0.1754	0.5335	0.6252
c.v.	0.0794	0.1626	0.1642

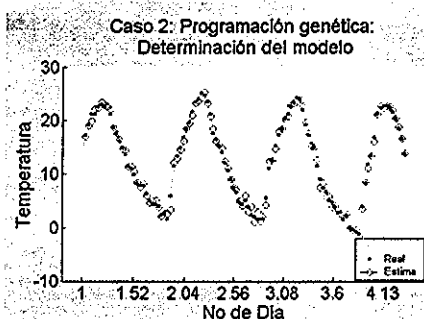


Figura 5.11: Estimación sobre las observaciones del grupo 1.

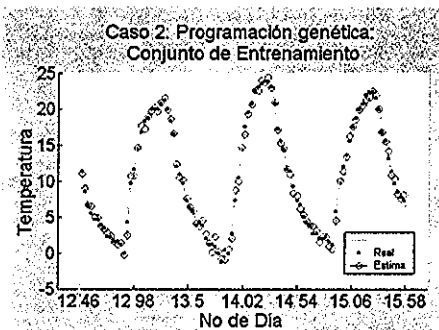


Figura 5.12: Estimación sobre las observaciones de entrenamiento (grupo 2).

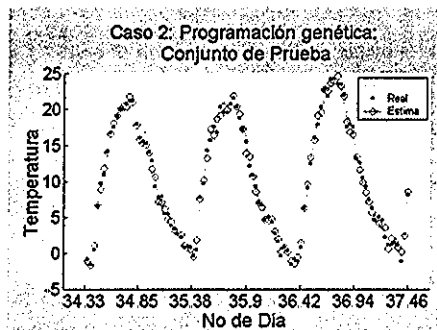
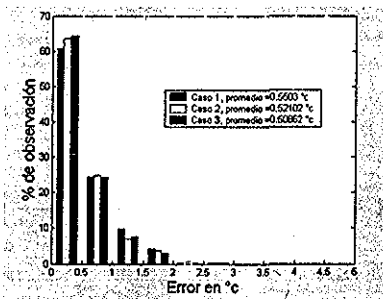


Figura 5.13: Estimación sobre las observaciones de prueba (grupo 3).

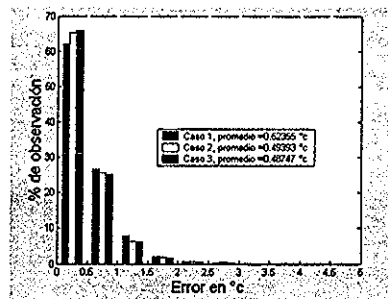
Para el primer caso, la ecuación [5 1] representa al mejor modelo, utilizando básicamente la temperatura. De las variables utilizadas por los mejores modelos de cada una de las 20 corridas el 31% de estas variables fueron temperatura. Sin embargo, el 42.5% fueron humedad relativa y el 17.5% fue radiación solar. A pesar de que se determinaron modelos que incluían la humedad relativa y la radiación solar, éstos no presentaron el mejor resultado. Las variables de temperatura, humedad relativa y radiación solar son las que tienen los coeficientes de correlación más altos con

respecto a la variable de temperatura en el siguiente tiempo. Para el segundo caso, el mejor modelo [5.3] está compuesto por temperatura y radiación solar. La radiación solar es importante pero no tiene efecto sobre las mediciones y estimaciones que se realizaron durante la noche. Los otros modelos consideran la dirección del viento, en un porcentaje similar al de la radiación solar (18% aproximadamente). Sin embargo, los coeficientes de correlación de la dirección del viento con respecto a la variable de temperatura en el siguiente tiempo son mucho menores. Para el caso 3, la ecuación [5.5] considera las variables de temperatura, humedad relativa, radiación solar y dirección del viento. Los otros modelos resultado de las 20 corridas también proponen modelos con combinaciones de estas cuatro variables.

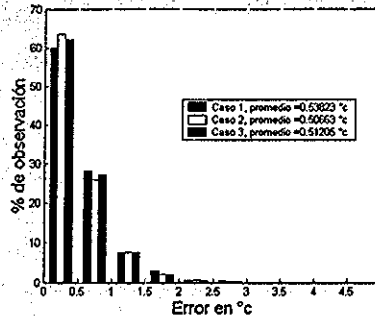
En cuanto a las Tablas 5.1, 5.5 y 5.9, se aprecia que si la función de evaluación se define por medio de la norma 1, los valores de la norma infinito crecían. Asimismo, cuando se evaluó bajo la norma infinito, los valores de la norma 1 son más altos. El comportamiento de la distribución de los errores es que las evaluaciones con norma infinito reducen el rango de errores pero las estimaciones son menos precisas que las evaluaciones hechas con norma 1. Las estimaciones resultado de modelos obtenidos al evaluar la norma 1 tienden a ajustarse más a los valores reales pero existen puntos donde los errores son altos. Para el caso 3, el comportamiento es en promedio un desempeño intermedio entre los casos 1 y 2, aunque en realidad presenta soluciones con buenas métricas infinito, otras con buenas métricas 1 y casos intermedios. Las gráficas para estos casos muestran un comportamiento semejante, siguen el fenómeno real. En las Figuras 5.14, 5.15 y 5.16, se presentan las distribuciones del error absoluto obtenido con los modelos de las ecuaciones [5.1], [5.3] y [5.5], para el caso 1, 2 y 3, respectivamente con PLP.



Figuras 5.14: Distribución de los errores para el conjunto 1.
No. de observaciones: 323



Figuras 5.15: Distribución de los errores para el conjunto 2.
No. de observaciones: 2162



Figuras 5.16: Distribución de los errores para el grupo 3.
No de observaciones: 1441

Como se observa, el comportamiento es muy semejante y con diferencias poco significativas como se observa en el análisis de las 20 corridas que se realizó posteriormente.

5.5 Caso 4:

El resultado de las diferentes corridas del algoritmo genético se presentan en la Tabla 5.13, donde se muestran los parámetros de vigilancia y los pesos correspondientes para cada una de las variables de entrada. Los resultados de cada corrida se muestran en el anexo J. En el anexo K se presentan los errores obtenidos tanto en la fase de entrenamiento como en la fase de prueba. Los resultados de las corridas se resumen en la Tabla 5.14. En las Figuras 5.17 y 5.18 se manifiestan las estimaciones realizadas comparándolas con los valores reales. El caso 4 considera los datos a escala real y predice directamente la siguiente temperatura.

Tabla 5.13: Parámetros determinados por algoritmos genéticos.

Corrida	Parámetro Vigilancia	Parámetro Vigilancia	Peso 1	Peso 2	Peso 3	Peso 4	Peso 5	Peso 6
Mín.	0.22954	0.96126	0.25913	0.00069662	0.0016922	0.00063334	0.020664	0.0009269
Prom.	0.37223	0.98778	0.66292	0.031769	0.035886	0.0047126	0.099791	0.16492
Máx.	0.82019	0.999	0.95444	0.14946	0.21344	0.02883	0.25148	0.39622
Desv.	0.15602	0.013035	0.18515	0.040922	0.065685	0.0081065	0.057285	0.11534
c.v.	0.19023	0.013048	0.19399	0.27379	0.30775	0.28118	0.22779	0.29111
Mejor	0.4191	0.0132	0.2793	1.2881	1.8304	1.7202	0.5740	0.6994

Tabla 5.14: Errores obtenidos durante la fase de entrenamiento y de prueba.

Corrida	Entrena		Prueba		No. de Patrones	Tiempo estimación	Tiempo para A.G.
	Norma 2	Norma infinito	Norma 2	Norma infinito			
Min.	23.485	3.9	98.876	7.5	347	0.29833	37.564
Prom.	138.11	10.364	142.75	10.9	858.25	0.65701	78.47
Más.	410.86	19.549	341.78	19.549	1733	1.1425	132.7
Desv.	92.073	3.8044	63.156	3.1005	363.57	0.24188	39.1
c.v.	0.6667	0.3671	0.4424	0.2844	0.4236	0.3682	0.4983

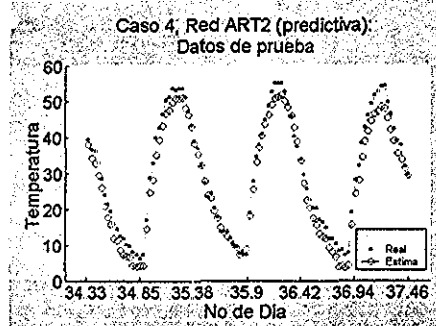
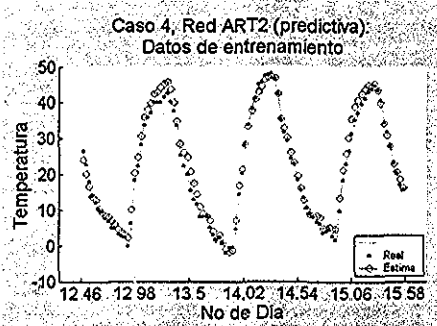


Figura 5.17: Estimaciones obtenidas durante la fase de entrenamiento (°C).

Figura 5.18: Estimaciones obtenidas durante la fase de prueba (°C).

5.6 Caso 5:

En el caso 5 se utilizaron los datos normalizados para la predicción en el incremento de temperatura. Este incremento se utiliza para determinar la siguiente temperatura. Una síntesis del resultado del algoritmo genético se exhibe en la Tabla 5 15, mientras que los mejores resultados de cada corrida se muestran en el anexo L

Tabla 5.15: Parámetros determinados por algoritmos genéticos.

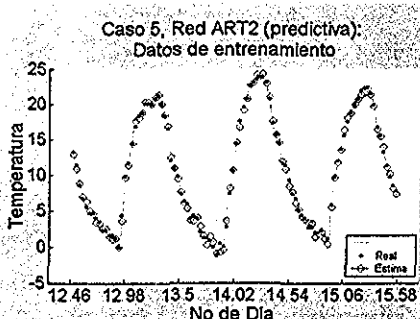
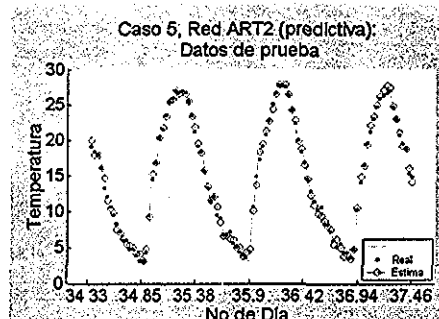
Corrida	Parámetro Vigilancia	Parámetro Vigilancia	Peso 1	Peso 2	Peso 3	Peso 4	Peso 5	Peso 6
Min.	0.018499	0.091007	0.0010779	0.012381	0.14617	0.00085375	0.062973	0.012351
Prom.	0.034386	0.5395	0.12216	0.16053	0.24085	0.074211	0.18999	0.21227
Más.	0.064633	0.9358	0.25205	0.2887	0.38735	0.15979	0.3837	0.34443
Desv.	0.011042	0.24719	0.068007	0.075124	0.064303	0.046277	0.090061	0.07807
c.v.	0.3211	0.4582	0.5567	0.4680	0.2670	0.6236	0.4740	0.3678
Mejor	0.046359	0.48445	0.073091	0.085048	0.26494	0.0062252	0.32768	0.24301

La Tabla 5.16 presenta los errores tanto para la fase de entrenamiento y la fase de prueba, detallándose en el anexo M los resultados por corrida En las Figuras 5.18 y 5.19 se comparan los valores estimados contra los valores reales.



Tabla 5.16: Errores obtenidos durante la fase de entrenamiento y de prueba.

Corrida	Entrena		Prueba		No. de Patrones	Tiempo estimación	Tiempo para A.G.
	Norma 2	Norma infinito	Norma 2	Norma infinito			
Min.	17.741	1.9	18.336	2.4375	160	0.19683	29.363
Prom.	23.403	3.2719	22.194	3.027	575.15	0.3881	62.891
Máx.	30.035	4.4	25.436	3.7	1078	0.61417	102.36
Desv.	3.4387	0.74313	2.2618	0.43116	278.81	0.12157	33.094
c.v.	0.1469	0.2271	0.1019	0.1424	0.4848	0.3132	0.5262

Figura 5.18: Estimaciones obtenidas durante la fase de entrenamiento ($^{\circ}\text{C}$).Figura 5.19: Estimaciones obtenidas durante la fase de prueba ($^{\circ}\text{C}$).

5.7 Caso 6:

En el último caso utilizando la red ART2, los datos se utilizaron en las escalas reales y se predijo el incremento de temperatura entre dos mediciones. Resumen de los resultados de las corridas del algoritmo genético y sus errores se presentan en las Tablas 5.17 y 5.18, respectivamente. Los resultados de cada una de las corridas realizadas se presentan en el anexo N para los resultados del algoritmo genético y en el anexo O para el desempeño

Tabla 5.17: Parámetros determinados por algoritmos genéticos.

Corrida	Parámetro Vigilancia	Parámetro Vigilancia	Peso 1	Peso 2	Peso 3	Peso 4	Peso 5	Peso 6
Min.	0.3239	0.96639	0.064406	0.0013564	0.0030755	0.0004888	0.032151	0.041509
Prom.	0.62606	0.9941	0.23611	0.21032	0.13469	0.014992	0.19133	0.21256
Máx.	0.99631	0.99899	0.5266	0.34885	0.30371	0.033289	0.3606	0.42268
Desv.	0.24417	0.0076575	0.10322	0.1071	0.098477	0.011334	0.098327	0.1141
c.v.	0.3900	0.0077	0.4372	0.5092	0.7311	0.7560	0.5139	0.5368
Mejor	0.45597	0.99688	0.34123	0.0013564	0.0030755	0.00062219	0.26712	0.38659

Tabla 5.18: Errores obtenidos durante la fase de entrenamiento y de prueba.

Corrida	Entrena		Prueba		No. de Patrones	Tiempo estimación	Tiempo para A.G.
	Norma 2	Norma infinito	Norma 2	Norma infinito			
Min.	12.436	1.5667	19.12	2.4958	191	0.41933	43.919
Prom.	24.659	2.8226	24.834	3.1548	991.55	1.1779	77.942
Máx.	47.047	4	37.787	4.7	1661	1.8337	129.94
Desv.	7.084	0.6958	3.7933	0.43498	622.8	0.60221	30.548
c.v.	0.2873	0.2465	0.1527	0.1379	0.6281	0.5113	0.3919

La Figura 5.20 y 5.21 muestran las estimaciones comparadas con los valores esperados para la fase de entrenamiento y la de prueba, respectivamente

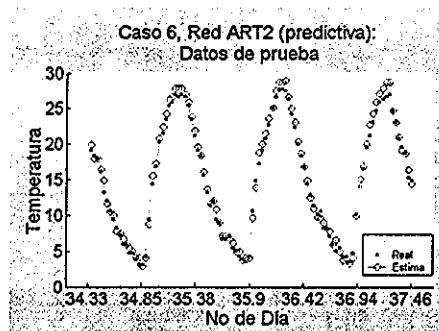
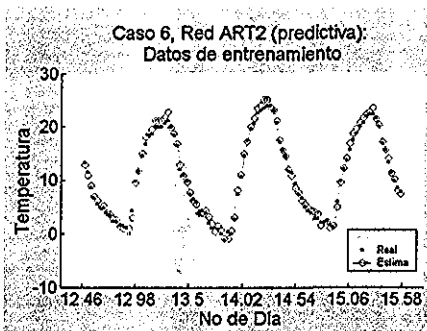


Figura 5.20: Estimaciones obtenidas durante la fase de entrenamiento (°C).

Figura 5.21: Estimaciones obtenidas durante la fase de prueba (°C).

Las variables de más peso para el caso 4 fueron la temperatura y la hora del día, resultados semejantes al caso 1 el cual usa programación genética. Para el caso 5, se considera la radiación solar, la dirección del viento y la hora del día, resaltando la poca influencia de la temperatura. En el caso 6, resalta la contribución de la temperatura y la hora del día como el caso 5, pero también considera la radiación solar. Como se observa en las Tablas 5.14, 5.16 y 5.18, el mejor desempeño durante la fase de entrenamiento se obtuvo para el caso 6, pero en el proceso de prueba el caso 5 sobresalió. Este hecho se debe en parte a la poca importancia que le da a la temperatura, y en esta etapa se utiliza la temperatura estimada para predecir el siguiente valor. Las estimaciones fueron mejores con el incremento de temperatura y utilizando la misma para predecir la siguiente temperatura en lugar de predecir directamente la temperatura siguiente (para $t+1$).

Las Figuras 5.22 y 5.23 muestran las distribuciones del error tanto para el caso de entrenamiento y de prueba, ambas tienen un comportamiento semejante. También se aprecia que la diferencia fuerte es entre el caso 4 y los otros dos casos.

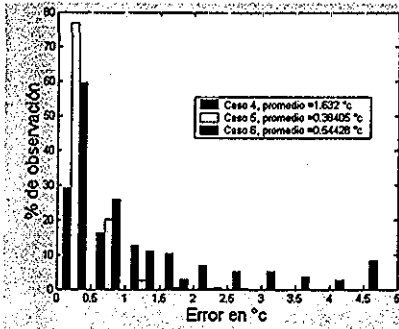


Figura 5.22: Grupo de entrenamiento.

No. de observaciones: 2164

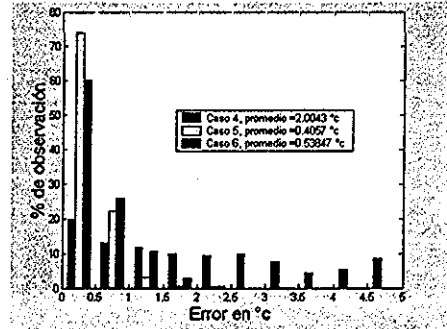


Figura 5.23: Grupo de prueba.

No. de observaciones: 1443

5.8 Caso 7:

El último caso utiliza un sistema borroso en el cual el antecedente es determinado con algoritmos genéticos. Los datos se normalizaron y se predijo el incremento de temperatura que ocurría entre una medición y otra. La Tabla 5.19 muestra la norma 2 y la norma infinito para cada una de las fases de determinación del sistema. El resultado de las mejores propuestas de cada corrida se presentan en el anexo P. Las Figuras 5.24, 5.25 y 5.26 detallan la distribución del error absoluto y las Figuras 5.27, 5.28 y 5.29 muestran las estimaciones a corto plazo. La Figura 5.30 y la Figura 5.31 muestran el resultado para PLP. En la Tabla 5.20 se muestra el número de casos, por corrida, en los que no fue posible realizar la siguiente estimación con los valores calculados anteriormente por estar dichos valores fuera de rango.

Tabla 5.19: Caso 7. Errores obtenidos durante la fase de entrenamiento.

Corrida	Obtención de Coeficientes		Ajuste de coeficientes		Prueba		Tiempo AG
	Norma 2	Norma infinito	Norma 2	Norma infinito	Norma 2	Norma infinito	
Mínimo	26.39	2.46	23.15	3.003	22.14	2.731	12.86
Mediana	45.65	3.398	93.91	24.19	40.35	4.974	14.14
Máximo	97.56	9.719	2455	824.4	1256	1073	15.5
Desviación	19.98	2.043	550	184	271.8	238.7	0.8594
c.v.	0.4377	0.6012	5.8567	7.6064	6.7361	47.9895	0.0608
Mejor	31.78	2.46	42.8	3.363	24.42	3.776	15.07

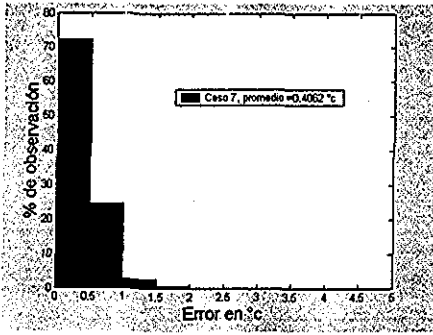


Figura 5.24: Grupo de creación.
No. de observaciones: 324

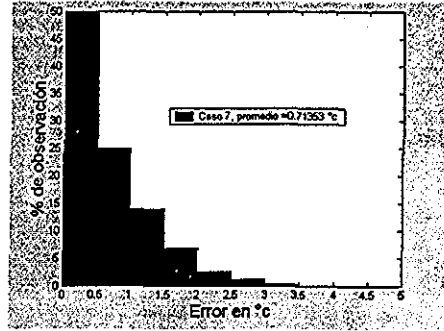


Figura 5.25: Grupo de entrenamiento.
No. de observaciones: 2164

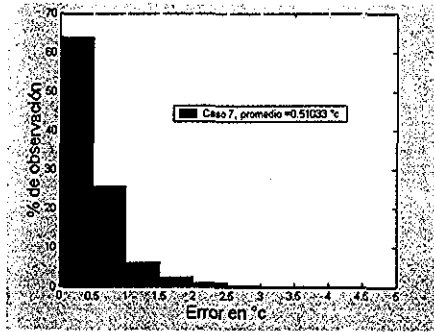


Figura 5.26: Grupo de prueba.
No. de observaciones: 1443

TESIS CON
FALIA DE ORIGEN

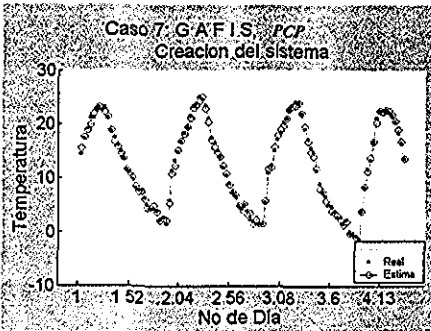


Figura 5.27: Estimación cuando se genera el FIS.

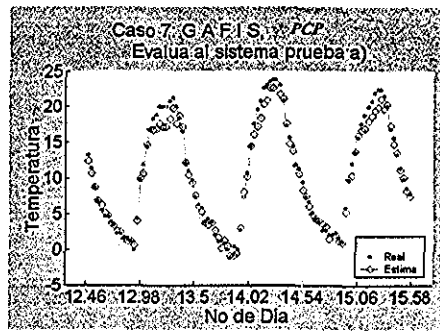


Figura 5.28: Estimación durante entrenamiento.

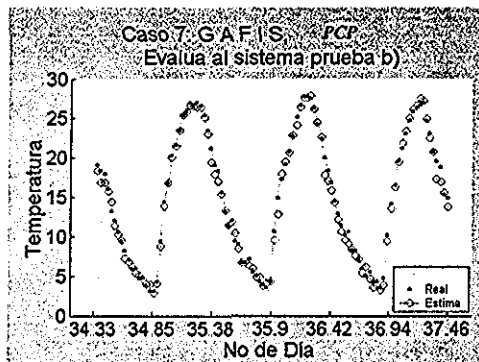


Figura 5.29: Estimación durante prueba

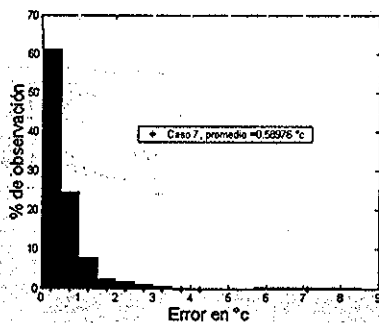


Tabla 5.30: Predicción a largo plazo

No. de observaciones: 1443

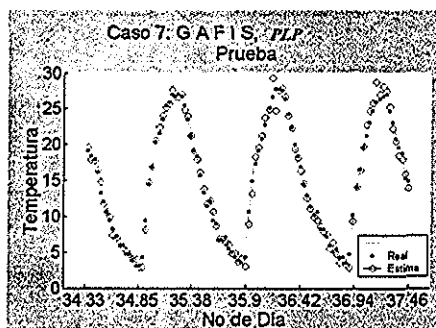


Figura 5.31: Prueba de extrapolación del sistema.

Tabla 5.20: No. de estimaciones que no pudieron activar ninguna regla.

Corrida	No. de Fallos	Corrida	No. de Fallos
1	18	11	40
2	117	12	41
3	45	13	50
4	12	14	110
5	112	15	169
6	5	16	7
7	61	17	150
8	17	18	38
9	25	19	39
10	19	20	10
		Mínimo	5
		Media	43.1
		Máximo	117
		Desviación	41.05673581
		C.V.	0.952592478

Los datos de entrada requirieron de un pre-proceso dado que las matrices resultantes para la estimación de los parámetros del consecuente estaban mal condicionadas, por lo que las estimaciones eran erróneas. En parte se debe al poco uso de variables, pero un incremento en el número de variables o en el número de funciones de pertenencia incrementa exponencialmente el uso de recursos y tiempo, lo cual hace el proceso prohibitivo. Por ejemplo, para polinomios lineales, si se tienen 3 variables con 3 funciones de pertenencia para cada variable se requiere calcular 108 parámetros, con 3 variables y 4 funciones de pertenencia se requieren 256 parámetros y con 4 variables y 3 funciones de pertenencia se requiere calcular 405 parámetros. Si el número de reglas se incrementa demasiado, la descripción del sistema en términos lingüísticos se dificulta.

5.9 Comparación en cuanto a precisión

La comparación entre los métodos se muestra en la Figura 5.32. Se observa que la red ART2 obtuvo un mejor desempeño, pero también que la diferencia no es muy grande, a excepción del GAFIS que puede tener casos con errores más grandes en unos pocos puntos.

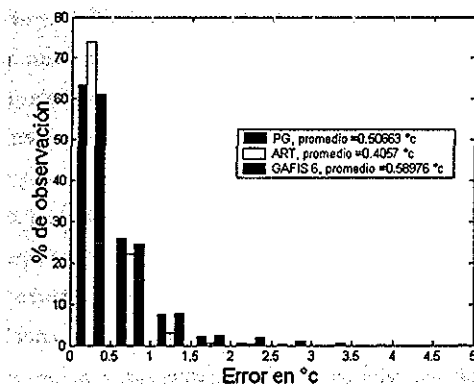


Figura 5.32: Comparación entre las tres técnicas

La técnica que utilizó GAFIS está dentro del error promedio, pero es inestable para realizar estimaciones. Debido a ello no se compara con las otras técnicas en el siguiente capítulo. En el siguiente capítulo se compara las técnicas de PG y ART en términos de complejidad de soluciones y la precisión de las mismas, considerando todas las corridas.

Capítulo 6

6 Discusión, Conclusiones y Trabajos Futuros

6.1 Discusión

El primer aspecto comparativo es el poder de cada técnica para predecir sobre puntos desconocidos. La Figura 6.1 muestra la capacidad predictiva que puede tener cada técnica correspondientes a sistemas con dos variables independientes (x,y) . Los puntos de soporte son marcados con *, la parte sombreada es el espacio donde cada técnica puede regresar un valor estimado. Los puntos con asterisco son los lugares donde sabemos realmente como se comporta el fenómeno, suponiendo que no hay error de medición. En todos los demás puntos solo podemos estimar lo que sucede. Las técnicas tradicionales asignan a priori propiedades las cuales debe tener el comportamiento del fenómeno entre dos o más puntos. Sobre la base de esas propiedades se realiza la estimación del valor de la variable dependiente en los puntos sin observación. Debido a que se requiere al menos dos puntos, pueden existir puntos (x,y) en los cuales no pueden realizarse estimaciones. Para el caso de la técnica de programación genética, dicha técnica puede encontrar una ecuación de la forma $z = f(x,y)$ siendo z el valor estimado. La función f puede realizar estimaciones en cualquier punto (x,y) donde la ecuación pueda ser evaluada. Para la técnica GAFIS, la estimación puede realizarse dentro de los rangos de las variables independientes. Es requisito indispensable que para poder proponer una estimación del punto a evaluar, los valores de todas las variables independientes estén dentro del rango con el cual fue entrenado el sistema borroso. Para el caso de la red ART, la capacidad de predicción se ubica en la vecindad del patrón. Si un vector de entrada (variables independientes) es vecino cercano o no, depende del valor de relación de semejanza y del parámetro de vigilancia. Si un vector de entrada no pertenece a ningún patrón, es un punto que corresponde a la parte clara de la Figura 6.1 correspondiente a la red ART2, la parte no predecible por la ART. Si bien con modificaciones del proceso puede predecir espacios semejantes a los que se obtienen con métodos tradicionales o por FIS, las modificaciones se basan en considerar varios grupos de patrones y no uno solo.

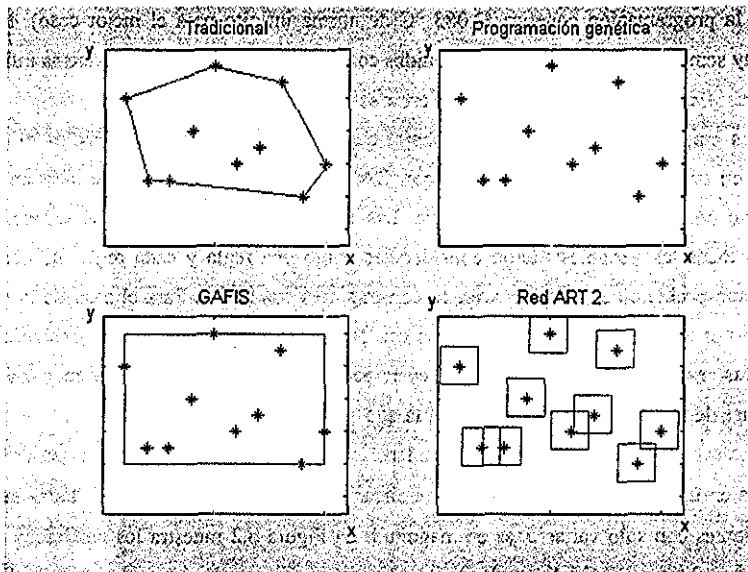


Figura 6.1: Comparativa entre técnicas para estimar valores fuera de los rangos de los datos conocidos

Para el caso de GAFIS, ésta fue la única técnica que localizó vectores de entrada de variables independientes en los cuales no fue posible hacer estimaciones de la variable dependiente. Esto fue debido a que para algunos casos, el valor estimado quedaba fuera del rango aceptable del FIS y dichos valores no pudieron usarse para predecir la siguiente estimación. Este caso corresponde a la parte clara, no predecible, de la Figura 6.1 que corresponde al GAFIS. Este resultado es un indicador de que las probabilidades de encontrar estos puntos es más alta para el GAFIS. Como se mencionará a detalle posteriormente, hay un parecido entre el caso que utiliza el ART y el caso que utiliza un GAFIS, por lo que sería interesante preguntarnos si este problema también es alto cuando se utiliza la red ART. Las ecuaciones obtenidas con programación genética al parecer son menos sensibles a este problema, inclusive se puede considerar en la función objetivo castigar el *fitness* si el modelo es no continuo.

El otro aspecto comparativo es la precisión de las estimaciones. El reto más difícil es encontrar una medida que verifique el valor de la estimación sin conocer el valor real; para este trabajo, la medida de la precisión es una medida pragmática. Qué tan buena o mala puede ser una estimación depende de las nuevas mediciones. El error obtenido durante el proceso de prueba es el valor que refleja más la precisión de las estimaciones, ya que no está considerando la mejora por sobre-entrenamiento, sino únicamente su poder de generalización. Las mejores estimaciones fueron las obtenidas por

TESIS CON
FALLA DE ORIGEN

ESTA TESIS NO SALE
DE LA BIBLIOTECA

medio de la programación genética (2 083 °C de norma infinito para el mejor caso), las cuales fueron muy semejantes a los resultados obtenidos con la red ARI (2 826 °C de norma infinito para el mejor caso), inclusive sus histogramas de error son semejantes

Uno de los puntos importantes en este trabajo, que fue común en dos técnicas, es el dividir el problema en sub-problemas. Para el caso con GAFIS, se divide cada variable independiente en rangos, uno por cada función de pertenencia. Las diferentes combinaciones que se pueden hacer utilizando todas las variables fueron consideradas como una regla y cada regla fue considerada como un sub-problema el cual fue resuelto con mínimos cuadrados. Para el caso de la red ART, cada patrón es un sub-problema. Para el caso con programación genética, no se consideró ninguna división, las pruebas realizadas con divisiones no generaron beneficio. Sin embargo, pueden existir otras formas de dividir el espacio del problema que no se ha explorado.

Para programación genética la meta era considerar los efectos de diferentes normas para evaluar los individuos en un problema de regresión simbólica. En teoría, en un espacio \mathbb{R}^n , todas las normas son semejantes con solo variaciones en magnitud. La Figura 6.2 muestra los resultados de las 20 corridas para obtener el modelo, evaluadas por la norma uno para los casos 1, 2 y 3. La Figura 6.3 muestra los resultados bajo la norma 2 y los resultados con la norma infinito se muestran en la Figura 6.4. En las Figuras 6.2 y 6.3 se muestra que las normas 1 y 2 califican igual a los individuos. Para comparar las gráficas de las Figuras 6.2 y 6.3 se escalaron los datos en el rango [0,1] y se aplicó un análisis de variancia en el cual no se encontró diferencia significativa para ninguno de los casos ($F_{(0,9138)}=2.842$, las F calculadas para el caso 1 fue de 0.280, caso 2 de 0.004 y para el caso 3 de 0.010). Sin embargo, este comportamiento no ocurre para las mediciones entre la norma infinito y las otras dos normas. Al comparar los valores, la norma 1 y la norma infinito tienden a tener efectos contrarios. Por un lado, la norma 1 tiende a ajustar más a los puntos de soporte pero hay casos aislados que tienden a tener diferencias grandes; mientras que con la norma infinito se evitan estos casos, pero en general el error con respecto a los puntos de soporte tiende a ser de mayor magnitud. Este efecto contrario se puede apreciar en la Figura 6.2 donde los mejores resultados se obtienen cuando se aplicó la norma 1 en la función objetivo, después cuando la función objetivo consideró tanto la norma 1 como la norma infinito y por último los casos que fueron obtenidos cuando se utilizó solamente la norma infinito como función objetivo. Por otra parte, la Figura 6.4 mostró un comportamiento completamente inverso. Mientras mayor precisión se obtenga en las soluciones este efecto contrario tiende a desaparecer, el caso extremo se presenta cuando la norma infinito es 0, la norma uno también es 0. Dentro de las posibles decisiones que se pueden considerar son: cuando las soluciones son aceptables aún con la presencia de este efecto contrario, entonces puede ser importante pensar que norma utilizar. Sin embargo, las soluciones obtenidas por

los tres casos generan un conjunto de soluciones no dominadas (frente de Pareto) por lo que la decisión de cuál caso utilizar esta en función de otros criterios basados en los objetivos del que tome la decisión. Una segunda opción es buscar modificaciones en el código que mejoren la precisión de los resultados, aún con la posibilidad de que éstas no existan. La tercera y última opción es considerar otras medidas de evaluación como son el criterio de AIC, la medición de cuánta variación explica el modelo o definir alguna otra métrica

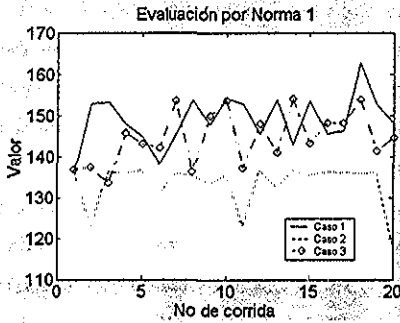


Figura 6.2: Evaluación de modelos por la norma 1.

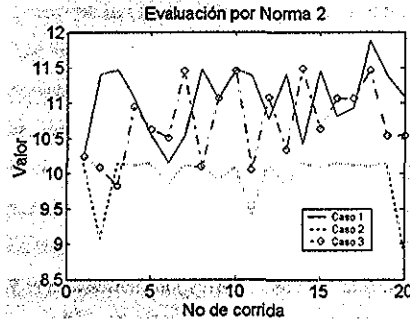


Figura 6.3: Evaluación de modelos por la norma 2.

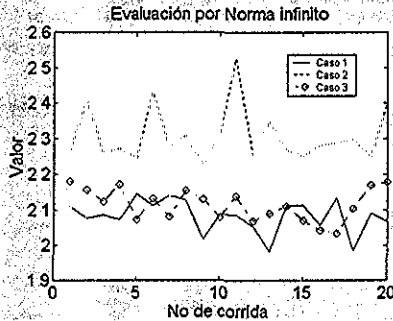


Figura 6.4: Evaluación de modelos por la norma infinito.

En cuanto a la predicción a largo plazo, el mejor desempeño en la etapa de prueba lo obtuvieron los casos donde se manejó la norma 1 en la función objetivo. Sin embargo, al realizar el análisis de variancia con un $\alpha = 0.10$ no se encontró diferencia significativa entre los casos y tampoco en la forma de evaluación para la comparación ($F_{(0.9, 2, 57)} = 2.398$, siendo la F calculada de 0.27 cuando se comparó bajo la norma 1, 0.994 bajo la norma 2 y para la norma infinito fue de 1.331). Por lo

TESIS CON
 FALLA DE ORIGEN

tanto, se observa que para predicciones a largo plazo, no se afecta en base a la norma seleccionada en la función objetivo.

En cuanto a la complejidad de los modelos obtenidos, no se encontró diferencia significativa entre los diferentes casos, el número de nodos por expresión estuvo en el rango de [10, 20]. Para el análisis de variancia se consideró un nivel de confianza del 10%, para una F de tablas de $F_{(0,9; 2, 57)} = 2.398$ y una F calculada de 0.201

La meta en la red ART fueron en dos sentidos: el primero evaluó la diferencia entre la predicción de la temperatura directamente comparado con la predicción del incremento de temperatura. La segunda comparación resultó entre utilizar los datos escalados entre [0,1] y el no aplicar dicho escalamiento de los datos. La tendencias del comportamiento de las normas 2 y la norma infinito para evaluar el desempeño de la red, fueron semejante aunque con diferencias en magnitud. Un ejemplo de estas gráficas se muestran en la Figura 6.5 y la Figura 6.6, donde se presenta los resultados de la norma 2 y la norma infinito para el caso 4, respectivamente. Para determinar si hay evidencia estadística se efectuó un análisis de variancia con un nivel de confianza del 10%, para cada uno de los casos correspondientes a la red ART (4, 5 y 6). Los datos fueron escalados en un rango de [0, 1] para evitar el efecto de la diferencia de magnitudes. En estos casos, la F de tablas fue de $F_{(0,9; 1, 38)} = 2.842$ con una F calculada de 1.541, 0.536 y 0.013 para los casos 4, 5 y 6, respectivamente. Dado lo anterior se presenta las comparaciones solamente utilizando las medidas de la norma infinito

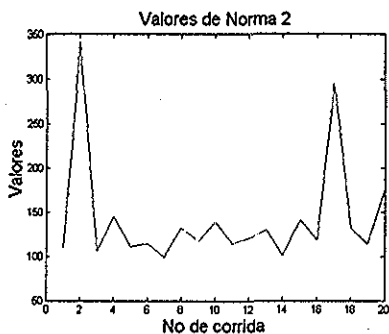


Figura 6.5: Valoración de Norma 2 de las 20 corridas caso 4.

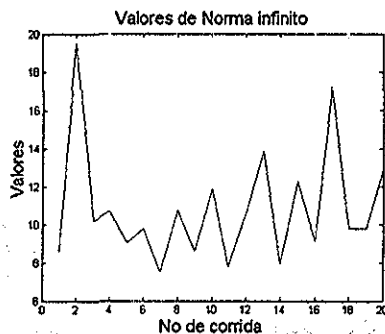


Figura 6.6: Valoración de Norma infinito de las 20 corridas caso 4.

En cuanto a la primera comparación, se utilizó la información del caso 4 y del caso 6 bajo la norma infinito para medir el desempeño de la red. La diferencia entre la predicción directamente de la temperatura y la predicción de temperatura en base al incremento de la misma es altamente significativa. El análisis de variancia obtuvo una F calculada de 122.387 comparada con una

$F_{(0,9 | 38)} = 2.842$. Para el segundo caso considerando como fijo la predicción del incremento de temperatura, se compararon el caso 5 (escalando los valores en un rango de [0,1]) y el caso 6 (sin escalamiento). El resultado del análisis de variancia no mostró evidencia estadística para rechazar la hipótesis de comportamientos diferentes. El nivel de confianza utilizado en dicha prueba fue de 10%, con $F_{(0,9 | 38)} = 2.842$ y una F calculada de 0.870

También se analizó el efecto que tuvo, por una parte, el seleccionar el criterio a predecir y, por otra, el efecto de escalar los datos sobre el número de patrones que se generaban. En cuanto al criterio de predecir, no se encontró la evidencia suficiente para determinar que criterio emplear, siendo la F calculada de 0.683. Sin embargo, el efecto de escalar los datos sí muestra ser significativo, con una F calculada de 7.448. En ambos análisis, la F de tablas tuvo un valor de $F_{(0,9 | 38)} = 2.842$ con un α del 10%

El resultado deseado para estas corridas era que ni el criterio para predecir, ni el escalamiento tuvieran efectos sobre la precisión de los resultados o sobre el número de patrones. La razón principal de que este resultado sea el esperado, es buscar técnicas que en primer lugar tengan un buen desempeño, pero también dependan menos de las decisiones conscientes o inconscientes de los usuarios. Lo que indican estos resultados es la importancia de considerar el fenómeno a predecir y que el escalamiento es recomendable para disminuir el costo computacional pero no absolutamente necesario.

La técnica del sistema de inferencia borroso fue difícil de manejar debido principalmente a que la matriz que resuelve los coeficientes que se calculan en el consecuente resultaba ser inestable y esto afectó directamente su desempeño. Esto nos lleva a concluir que se requiere de la investigación de métodos que puedan resaltar las bondades del uso de la lógica borrosa. El resultado de la técnica de predicción tradicional no tuvo resultados alentadores; sin embargo, pueden existir otras técnicas que tengan un desempeño semejante a los obtenidos por estas técnicas no convencionales.

En cuanto a la comparación entre técnicas, dicha comparación se realizó solo entre la red ART y la PG. En cuanto a complejidad, no se realizó ningún análisis de variancia dado la diferencia de magnitud. Para PG, las soluciones consideraron no más de 20 parámetros mientras que la red ART utilizó varias centenas de parámetros. En cuanto a la precisión de los resultados, no se encontró evidencia para determinar una diferencia en los resultados, la F calculada para este caso fue de 2.008 mientras que la F de tablas fue de $F_{(0,9 | 38)} = 2.842$ a un nivel de confianza del 10%.

Un aspecto importante de puntualizar son las semejanzas entre la técnica que utilizó un GAFIS y la técnica que utilizó una red ART, las cuales se resumen a continuación:

<i>GAFIS</i>	<i>ART 2</i>
<i>ANTECEDENTE</i>	<i>Pesos de la red</i>
<i>CONSECUENTE</i>	<i>Valor de estimación del patrón</i>
<i>FUNCIONES DE PERTENENCIA</i>	<i>relación de semejanza</i>
<i>REGLA</i>	<i>patrón</i>

Sin embargo, los resultados fueron distintos, lo cual hace pensar que son diferentes. Dichas diferencias también pudieron deberse a que el número de patrones fue más pequeño para GAFIS, con 27 patrones y con 575, 858 y 991 en promedio, dependiendo el caso para el ART. También influye el hecho de que la red ART no utiliza la información de otros puntos, sino únicamente el valor del patrón ganador.

6.2 Conclusiones

Las conclusiones generales de este trabajo se puntualizan a continuación.

- La programación genética puede detectar el modelo y las variables que mejor describan el fenómeno, considerando la menor complejidad.
- La representación de ramas para el caso de la PG, permite controlar más fácilmente el problema de crecimiento de los árboles o la complejidad de las soluciones en comparación con la representación tradicional
- Para el caso analizado en PG, las diferentes normas pueden tener efectos diferentes en los resultados obtenidos, sobre todo la norma 1 y la norma infinito que son los casos extremos
- Los mejores resultados fueron obtenidos por la red ART 2 y la programación genética.
- La implementación con red ART y GAFIS pueden conservar parte de su aprendizaje con la entrada de nueva información.
- El consecuente del GAFIS y su equivalente en la red ART 2 pueden ser precisados de una mejor manera mediante la programación genética (propuesta de una técnica híbrida), la cual no obstante podría proporcionar mejoras en las predicciones, incrementaría el tiempo de procesamiento
- La desventaja más fuerte de la red ART 2 es su imposibilidad para describir el fenómeno más allá de la estimación de valores de la variable dependiente

6.3 Trabajos Futuros

- a) Buscar formas de aceleración de convergencia utilizando programación genética. En específico, la implementación de un operador de cruce no aleatorio, donde la cruce se realice en forma dirigida sobre la base del aporte de varianza de cada rama al modelo.
- b) Aplicar otros métodos en la determinación de los parámetros del consecuente como son el gradiente conjugado.
- c) Buscar una forma de clasificar los patrones de tal manera que se pueda describir el fenómeno.
- d) Explotar la propuesta de representación en programación genética introducida en este trabajo. Esto es, extender a otros dominios y/o problemas.
- e) Analizar datos diversos (series) los cuales presenten diferentes características, con el fin de evaluar el alcance de la PG y otras técnicas.

Bibliografía

- [1] Babuška R. H B Verbruggen and H.J.L. van Can (1999) Fuzzy Modeling of Enzymatic Penicillin – G conversion. *In Engineering Application of Artificial Intelligence* Vol 12 (1) pp. 79-92
- [2] Bahat M., G. Inbar, O. Yaniv and M A. Schneider (2000) Fuzzy Irrigation Controller System *in Engineering Application of Artificial Intelligence*. Vol 13 (2) pp. 137 – 145.
- [3] Bonham C.R. and I.C. Parmee. (1999) Improving the performance of cluster oriented genetic algorithms (COGAs) *in Evolutionary Computation 1999 CEC 99 Proceedings of the 1999 Congress*. Vol 1, pp. 554-561
- [4] Burden R.L. and D.J. Faires. (1988). *Numerical Analysis* fourth edition. PWS-KENT Publishing Company Boston
- [5] Carpenter G., and S. Grossberg (1988) The ART or Adaptive Pattern Recognition by Self Organizing Neural Network *in Computer IEEE* March 1988, Vol. 21(3) pp. 77-88.
- [6] Cvetkovic, D. and I.C. Parmee. (1999) Genetic Algorithm Based Multi-Objective Optimization and Conceptual Engineering Design *in Evolutionary Computation. 1999 CEC 99 Proceedings of the 1999 Congress*. Vol 1 pp 29 -35
- [7] Chaiyaratana N. and A.M. Zalzala (1999). Hybridization of Neural Networks and Genetic Algorithms for Time Optimal Control *Evolutionary Computation 1999 CEC 99 Proceedings of the 1999 Congress*. Vol 1 pp 389 - 396
- [8] Chapra C.S. and Canale (1990) *Numerical Methods for Engineers* second edition. Mc Graw Hill International Edition 811p
- [9] Corno F. M.S. Reorda and G. Squillero. (1999) Verifying the Equivalence of Sequential Circuits with Genetic Algorithms *in Evolutionary Computation 1999 CEC 99 Proceedings of the 1999 Congress*, Vol . 2, pp. 1293 – 1298
- [10] Dash P.K., A.C. Liew, S. Rahman and S. Dash. (1985) Fuzzy and Neuro – Fuzzy Computing Models for Electric load Forecasting *Engineering Application in Artificial Intelligence*. Vol. 8 (4), pp. 423-433.
- [11] Emami M.R., A.A. Goldenberg and B.I. Turksen (2000). Fuzzy Logic Control of Dynamic System from Modeling to Design. *Engineering Application in Artificial Intelligence*. Vol. 13 (1) pp 47-69
- [12] Fonseca C.M. and P.J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization *in S. Forrest editor Proceeding of the Fifth International Conference in GA San Mateo, July CA: Morgan Kaufmann*
- [13] Goldberg D. (1989) *Genetic Algorithm in Search and Optimization and Machine Learning*. Reading, MA Addison Wesley
- [14] Hans M. (1999) Identification of Enhanced Fuzzy Models with Special Membership Functions and Fuzzy Rules Based. *in Engineering Application in Artificial Intelligence* Vol. 12 (3). pp. 309 – 319
- [15] Hilerá J.R. y V.J. Martínez. (2000) *Redes Neuronales Artificiales*. Alfaomega Madrid. España
- [16] Johansen I.A. and R. Murray-Smith (1997) The Operating Regime Approach to Nonlinear Modeling and Control *Charter 1*, pp. 3 – 72.
- [17] Jung R.J.S (1993) Anfis Adaptive Network based Fuzzy Inference Systems *in Man and Cybernetics*, Vol 23(3) May/June 1993 pp 665-685
- [18] Kashyap R.L. and R.A. Ranachandra (1976) *Dynamic Stochastic Models from Empirical Data Mathematics in Science and Engineering*. Vol. 122 New York E.U.A. pp 334

- [19] Kaufmann A. and M M Gupta. (1985). *Introduction To Fuzzy Arithmetic*. Van Nostrand Reinhold Company Inc New York E.U.A pp. 351
- [20] Khoo I.P., C.L. Ang and J Zhang (2000) A Fuzzy Based Genetic Approach to Diagnosis of Manufacturing System in *Engineering Application in Artificial Intelligence*. Vol. 13(3). pp. 303-310
- [21] Kolaric I and G. Ruderford (1974) *Time Series Forecasting Using Neural Networks* API. pp. 86-94 citeseer.nj.nec.com/75250.html.
- [22] Koza J.R (1992). *Genetic Programming*. The MIT press London England
- [23] Kuo R J (2000) Multisensor Integration for on line Tool Wear Estimation through Artificial Networks and Fuzzy Neural Networks. in *Engineering Application in Artificial Intelligence* Vol. 13(3) pp 249-261
- [24] Kubota N, S Hashimoto and F. Kojima (2001) Genetic Programming for Life-Time Learning of a Mobile Robot IFSA in *World Congress and 20th NAFIPS International Conference 2001 Joint 9th* Vol 4. pp 2422-2427.
- [25] Lawrence S., A.C. Isoi and C.L. Giles (2001) Noisy Time Series Prediction Using Symbolic Representation and Recurrent Neural Network Grammatical Inference in *Machine Learning* Vol 44(42) July/August. pp 161-183.
- [26] Lee G.Y (1999). Genetic Recursive Regression for Modeling and Forecasting Real - World Charter 17. *Advances in Genetic Programming* Vol. 3 edited by Lee Spector. William B Landon Una-May O'Reilly and Peter J. Angeline pp 378-401 MIT Press
- [27] Lee G.Y (2001) Time Series Perturbation by Genetic Programming in *Evolutionary Computation 2001 CEC 01 Proceedings of the 2001 Congress*. Vol. 1 pp. 403 - 409
- [28] Liao J., M.J. Er and J. Lin (2000). Application of System for the Automatic Generation of Fuzzy Neural Networks in *Engineering Application in Artificial Intelligence*. Vol. 8(3). pp. 293 -302
- [29] Linkens, D.A. and N.H. Okola (1993) A distributed genetic algorithm for multivariable fuzzy control. in *Genetic Algorithms for Control Systems Engineering, IEEE Colloquium*, pp 9/1 -9/3
- [30] Lorenz, N.E (1963) Deterministic Nonperiodic Flow in *Journal Atmospheric Science* Vol. 20, pp. 130-141
- [31] Ljung L. (1999) *System Identification. Theory for the User*, second edition Prentice Hall PIR, E.U.A pp 609
- [32] Metha K and S Bhattacharyya (1999) Combining Rules Learnt Using Genetic Algorithms for Financial Forecasting in *Evolutionary Computation 1999 CEC 99 Proceedings of the 1999 Congress*. Vol 2. pp. 1245-1252
- [33] Molina J.M., P. Isasi, A. Berlanga and A. Sanchis Hydroelectric Power Plant Management Relying on Neural Network and Expert System Integration *Intelligence Application in Engineering* 13 (2000) pp 357 - 369
- [34] Morimoto I., J. Susuki and Hashimoto (1997) Optimization of Fuzzy Controller for Fruit Storage Using Neural Network and Genetic Algorithms in *Engineering Application in Artificial Intelligence* Vol. 10(5). pp. 453- 461
- [35] Moody J (1994) *Economic Forecasting Challenges and Neural Network Solutions NATO ASI Series F: Springer - Verlag from Static's to Neural Networks and Pattern Recognition Applications* Cherkassky - Fredman - Wechlece editors
- [36] Mühlenbein H and Schlierkamp-Voosen Predictive Models for the Breeder Genetic Algorithm: I Continuous Parameter Optimization', *Evolutionary Computation*, Vol 1 No 1, pp 25-49, 1993.
- [37] Nakamura (1992) *Métodos Numéricos Aplicados con Software primera edición* Prentice Hall - Hispanoamerica S.A.
- [38] Nelles, O (2001) *Nonlinear system identification : from classical approaches to neural networks and fuzzy models* Berlin ; New York : Springer-Verlag pp 785
- [39] Nielsen H.R. (1988) Neurocomputing Picking the Human Brain in *IEEE Spectrum* march 1988. pp. 36-42

- [40] Pratihari D.K., K. Deb and A. Ghosh. Fuzzy – Genetic Algorithms and Mobile Robot Navigation Among Static Obstacles in *Evolutionary Computation 1999. CEC 99 Proceedings of the 1999 Congress*. Vol 1 pp 327 – 334
- [41] Plummer A.E. (2000). *Time Series Forecasting with Feed-Forward Neural Networks: Guidelines and Limitations* Laramie, Wyoming. Tesis Maestría de la Universidad de Wyoming
- [42] Prawda W.J. (1980). *Métodos y Modelos de Investigación de Operaciones Volumen II Modelos estocásticos* Ed. Limusa. México. D.F pp 1026
- [43] Tang Y. and W. Hsieh (2001) Coupling Neural Networks to Incomplete Dynamical Systems Via Variational Data Assimilation in *American Meteorological Society*. Vol. 129 pp. 818-834
- [44] Tang Y., W. Hsieh. B. Tang and K. Haines (2001) A Neural Network Atmospheric Model for Hybrid Coupled Modelling in *Climate Dynamics* Vol 17 pp. 445 – 455.
- [45] Takagi I. and M. Sugeno. (1985). Fuzzy Identification of Systems and Its Applications to Modeling and Control in *IEEE Transactions on Systems, Man and Cybernetics* Vol. 15(1). January/February 1985 pp 116-132
- [46] Tamaki H., E. Nishino and S. Abe. (1999) A Genetic Algorithm Approach to Multi – Objective Scheduling Problems with Earliness and Tardiness Penalties. in *Evolutionary Computation 1999 CEC 99 Proceedings of the 1999 Congress*. Vol 1 pp. 46-52
- [47] Thiessing F.M. and O. Vomberger. (1997) Forecasting Sales Using Neural Networks in *proceedings ICNN97 Houston Texas 9-12. June 1997*, Vol. 4. pp. 2123 –2125
- [48] Schaefer C.G. Jr (1999). Morphogenesis of Path Plan Sequences through Genetic Synthesis of L – System Productions in *Evolutionary Computation 1999 CEC 99 Proceedings of the 1999 Congress* Vol. 1 pp. 358 – 365.
- [49] Shaw K.J., A.L. Nortcliffe, M. Thompson, J. Love, P.J. Fleming and C.M. Fonseca (1999) Assessing the Performance of Multiobjective Genetic Algorithms for Optimization of a Batch Process Scheduling Problem in *Evolutionary Computation 1999 CEC 99 Proceedings of the 1999 Congress* Vol. 1 pp. 37 – 45
- [50] Soleng H.H. (1999) Oil Reservoir Production Forecasting with Uncertainty Estimation Using Genetic Algorithms in *Evolutionary Computation 1999 CEC 99. Proceedings of the 1999 Congress*, Vol 2, pp. 1217-1223.
- [51] Vasilev V.K., J.F. Miller and I.C. Forgyarty (1999). Digital Circuit Evolution and Fitness Landscapes in *Evolutionary Computation 1999 CEC 99 Proceedings of the 1999 Congress*. Vol. 2. pp. 1299- 1306
- [52] Whigham P.A. and P.F. Crapper (1999). Time Series Modeling Using Genetic Programming: An Application to Rainfall – Runoff Models. Chapter 5. *Advances in Genetic Programming*, Vol 3. edited by Lee Spector, William B. Landon, Una-May O'Reilly and Peter J. Angeline pp. 89-104. MIT Press.
- [53] Yamane I. (1979) *Estadística Tercera edición*. Harla México. D.F pp 190
- [54] Yang Y., W. Hsieh. B. Tang and A. Haines (2001) Neural Network Atmospheric Model for Hybrid Coupled Modelling in *Climate Dynamics* Vol. 17 pp. 445-455 Springer – Verlag 2001
- [55] Yang S.H., B.H. Chen and X.Z. Wang (2000) Neural Network Based Fault Using Immensurable Input in *Engineering Application in Artificial Intelligence*. Vol. 13 pp. 345-356
- [56] Yao X. (1999) Evolving Artificial Neural Networks in *Proceedings of the IEEE* Vol 87(9) pp. 1423 –1447
- [57] Yoshihara I., T. Aoyama and M. Yasunaga (2000) GP-Based Modeling Method for Time Series Prediction with Parameter Optimization and Node Alternation in *Evolutionary Computation 2000 CEC 00 Proceedings of the 2000 Congress* Vol 2 pp. 1475-1481.
- [58] Yu D.L., J.B. Gommen, and D. Williams. (2000) Neural Model Input Selection for MIMO Chemical Process in *Engineering Application in Artificial Intelligence* Vol. 13(1). pp. 15-23
- [59] Rodríguez V.K. (2001). Genetic Programming in Time Series Modelling: An Application to Meteorological Data in *Evolutionary Computation 2001 CEC 01 Proceedings of the 2001 Congress* Vol. 1, pp. 261 – 266

- [60] Srinivasan, D , A C Liew, C.S Chang (1994). Forecasting Daily Load Curves Using a Hybrid Fuzzy-Neural Approach. in *Generation, Transmission and Distribution IEE Proceedings* , Vol. 141 (6) , Nov. pp. 561 –567.
- [61] Zadeh, L A (1965) *Fuzzy Sets in Information and Control*. Vol. 8, pp. 338 – 353
- [62] Zimmermann, H.J. (1991) *Fuzzy Set Theory and its Application second edition* Kluwer Academic Publisher. Boston\Dordrecht\London
- [63] Zitang B.T , P. Ohm and H Mühlenbein (2001). Evolutionary Neural Trees for Modeling and Prediction Complex System. in *Engineering Application in Artificial Intelligence*. Vol. 13(1). pp. 473 – 483.

Anexos

Anexo A

Modelos obtenidos con programación genética, con función objetivo de la norma infinito.

(modelo $h_{r,3} h_{t,1} (-r_{t,2} r_{r,1}) (\cos r_{r,1}) (* h_{t,2} v_{r,2})$ 0 054975 -0.052811 -0.0093394 -0.18047 -2 4983e-006 0.0020442)
 (modelo $(\cos r_{r,1}) (\sin H) (\cos t_{r,2}) h_{r,2} h_{t,3}$ -0.26069 0.10919 0.062656 -0.098071 0.10589 -0.26305)
 (modelo $h_{r,3} (\cos r_{r,1}) (\sin v_{r,3}) (\sin r_{r,1}) h_{r,2}$ 0.1077 -0.25343 -0.053389 0.053758 -0.099562 -0.29338)
 (modelo $t_{r,1} (\cos r_{r,1}) (-t_{r,2} h_{r,1}) h_{r,2}$ 0.11058 -0.19959 -0.10713 -0.098961 -0.35792)
 (modelo $(\cos t_{r,1}) (-t_{r,1} t_{r,2}) h_{r,3} h_{r,2} (\tan d_{r,1})$ 0.041448 0.46642 0.056642 -0.05316 -0.0015922 -0.17011)
 (modelo $t_{r,1} (-t_{r,3} t_{r,1}) t_{r,3} (-t_{r,1} t_{r,3}) (\sin H)$ 15.7325 7.6939 -15.7414 -7.6439 0.10088 0.10258)
 (modelo $t_{r,1} h_{r,3} (\sin t_{r,1}) h_{r,2} t_{r,2}$ 0.47435 0.055843 0.051803 -0.051537 -0.47089 -0.25667)
 (modelo $h_{r,3} (\sin v_{r,1}) h_{r,2} (\cos r_{r,1}) t_{r,2}$ 0.10795 -0.048736 -0.10133 -0.25515 -0.005192 -0.14974)
 (modelo $(\sin r_{r,1}) h_{r,2} (\cos r_{r,1}) (+ h_{r,3} t_{r,1}) t_{r,2}$ 0.071653 -0.10135 -0.21328 0.10745 -0.11301 -0.1385)
 (modelo $t_{r,2} (\sin r_{r,1}) (\cos r_{r,1}) (\sin v_{r,3}) (-h_{r,2} h_{r,1})$ -0.026506 0.072373 -0.25872 -0.042664 -0.10327 0.44337)
 (modelo $h_{r,2} (\sin h)$ $h_{r,3} t_{r,3} (\cos r_{r,1})$ -0.098266 0.10425 0.10226 -0.015433 -0.28278 0.12469)
 (modelo $(\cos r_{r,1}) t_{r,1} (-h_{r,3} t_{r,2}) (+ h_{r,3} h_{r,2}) v_{r,2}$ -0.16765 0.18926 0.1936 -0.093482 0.00064867 -0.31577)
 (modelo $(\text{divd } h_{r,2} h_{r,1}) h_{r,2} (-r_{r,1} t_{r,3}) h_{r,3} h_{r,3}$ 59.2592 -0.1012 0.0020126 0.12503 0 -60.8454)
 (modelo $(\sin t_{r,1}) t_{r,1} t_{r,2} (+ t_{r,3} h_{r,2}) h_{r,3}$ 0.04528 0.4396 -0.38558 -0.052427 0.056214 -0.20859)
 (modelo $h_{r,2} (\cos r_{r,1}) t_{r,3} h_{r,3} (\sin v_{r,1})$ -0.10024 -0.26506 -0.013532 0.10477 -0.043769 0.062666)
 (modelo $t_{r,2} (+ h_{r,3} t_{r,1}) (+ h_{r,2} h_{r,1}) (+ h_{r,2} h_{r,1}) (\cos r_{r,1})$ -0.20121 0.19539 -0.095159 0 -0.18391 -0.095716)
 (modelo $(-h_{r,3} h_{r,1}) t_{r,3} (\sin t_{r,1}) (\cos r_{r,1}) t_{r,3}$ 0.071062 -0.021592 0.046617 -0.19819 0 0.36062)
 (modelo $(\text{divd } h_{r,1} h_{r,1}) (+ r_{r,1} t_{r,2}) h_{r,1} h_{r,3} (+ ht-2 ht-3)$ 63.5879 0.0021275 -0.02066 0.24642 -0.10258 -65.0966)
 (modelo $(\cos r_{r,1}) (-h_{r,3} h_{r,2}) t_{r,3} h_{r,3} (\sin h)$ -0.28067 0.098265 -0.015494 0.0039735 0.10171 0.1253)
 (modelo $t_{r,1} (-h_{r,3} t_{r,2}) (\cos r_{r,1}) (+ t_{r,1} h_{r,2})$ 0.21026 0.10753 -0.20027 -0.099341 -0.35754)

Anexo B:

Errores de las mejores propuestas de cada corrida Caso 1

Corrida	Norma 1	Norma 2	Norma Infinito	No Nodos
1	136.4907	10.2837	2.1106	17
2	152.9171	11.412	2.0773	15
3	153.3806	11.4679	2.0859	15
4	147.9506	11.0783	2.0728	13
5	144.8256	10.5373	2.1454	16
6	138.2192	10.1565	2.1119	17
7	145.3765	10.5238	2.1413	13
8	153.6081	11.4793	2.1277	14
9	147.7951	11.0584	2.0186	16
10	154.2207	11.4659	2.0875	17
11	152.7659	11.3908	2.0869	14
12	145.6238	10.7829	2.0536	17

13	153.7785	11.4108	1.9826	16
14	142.8585	10.4224	2.1123	15
15	153.5059	11.4517	2.1131	14
16	145.5478	10.8185	2.0574	19
17	146.2302	10.9364	2.1335	16
18	162.7383	11.889	1.9844	18
19	152.7733	11.3909	2.0905	16
20	147.9723	11.0784	2.0668	15

Anexo C

Diferentes normas obtenidas por grupo, en predicción a largo plazo, Caso 1

Corrida	Norma 1			Norma 2			Norma infinito		
	Grupo 1	Grupo 2	Grupo 3	Grupo 1	Grupo 2	Grupo 3	Grupo 1	Grupo 2	Grupo 3
	-15%	-60%	-40%	-15%	-60%	-40%	-15%	-60%	-40%
1	136.5	910.1	1118	10.28	25.9	38.77	2.111	2.837	4.803
2	153	1014	949	11.41	28.53	32.49	2.134	2.837	4.163
3	153.4	1011	943.3	11.47	28.71	32.44	2.086	2.914	4.255
4	151	1002	908.5	11.33	28.61	31.32	2.05	2.99	4.125
5	157.3	1069	774.6	11.64	30.85	26.89	2.218	3.95	3.165
6	170.5	1050	732.9	12.61	30.77	25.75	2.719	4.665	2.975
7	157.8	1071	765.8	11.67	30.97	26.71	2.196	4.062	3.216
8	153.5	1009	942.3	11.47	28.64	32.35	2.122	2.904	4.244
9	151	1007	903.8	11.3	28.68	31.13	2	3.029	4.125
10	153.7	1005	907.7	11.44	28.61	31.25	2.06	2.995	4.077
11	152.5	1004	921.3	11.37	28.38	31.49	2.057	2.969	4.106
12	151.2	1011	865.8	11.24	28.78	29.82	2.057	3.282	3.817
13	153.7	1094	906.9	11.41	30.43	32.09	1.979	3.199	4.372
14	156.1	1047	751.7	11.59	30.35	26.3	2.243	3.958	3.178
15	153.2	1005	924.3	11.44	28.54	31.71	2.086	2.944	4.155
16	152	1008	859.7	11.29	28.8	29.71	2.062	3.204	3.919
17	146.1	943.7	988	10.93	26.94	33.63	2.131	2.941	4.241
18	162.8	1122	912.3	11.89	31.43	33.51	1.984	3.391	4.671
19	152.5	1004	921.5	11.37	28.38	31.49	2.06	2.968	4.105
20	151.1	1003	909.2	11.33	28.62	31.34	2.048	2.998	4.127

Anexo D

Modelos obtenidos con programación genética con función objetivo Norma 1.

- (modelo $(-d_{r1} h_{r1}) h_{r1} t_{r3} t_{r1} (-t_{r1} t_{r1})$ 0.0018196 0.0046171 -0.060405 0.05747 0.31033 -0.10897)
- (modelo $(* r_{r1} t_{r2}) (* r_{r1} t_{r1}) r_{r1} (+ t_{r3} t_{r1}) t_{r2}$ 0.00087437 -0.00092556 0.066203 -0.038852 -0.024251 0.045344)
- (modelo $(\tan d_{r2}) (-t_{r1} t_{r1}) (\sin d_{r2}) t_{r1} (-t_{r1} t_{r1})$ -0.01081 0.37133 -0.031497 -0.010549 0.0.12992)
- (modelo $t_{r1} (\sin d_{r2}) t_{r1} (\tan t_{r2}) (\exp t_{r2})$ -0.36769 0.070209 0.35844 0.0010199 -1.2029e-012 0.11464)
- (modelo $(c_{r5} t_{r1}) d_{r2} (+ t_{r1} t_{r1}) t_{r3} (\tan d_{r1})$ 0.019388 -0.0034044 0.18049 -0.36988 -0.00019612 0.12115)
- (modelo $r_{r1} (-t_{r3} t_{r1}) (* r_{r2} t_{r1}) (-t_{r3} t_{r1}) (-t_{r3} t_{r1})$ 0.0027429 -0.25364 -0.00011291 0.0 -0.11359)
- (modelo $t_{r1} (\tan d_{r2}) d_{r2} d_{r1} (-t_{r1} t_{r1})$ -0.010533 -0.025988 -0.025301 0.024218 0.36665 0.12661)
- (modelo $v_{r1} t_{r1} (\exp d_{r2}) h_{r1} t_{r3}$ 0.00061959 0.35461 -3.9058e-012 0.002552 -0.36198 -0.14637)

(modelo $(+ t_{i,3} t_{i,1}) (+ t_{i,3} t_{i,1}) (- r_{i,2} r_{i,1}) (- r_{i,2} r_{i,1}) t_{i,3} 0.30161 0 -0.0059973 0 -0.60369 0.009193)$
 (modelo $d_{i,2} t_{i,1} (* r_{i,3} d_{i,1}) (+ d_{i,2} t_{i,3}) v_{i,2} 0.33706 0.34325 3 2424e-005 -0.3555 0.0003635 0.097718)$
 (modelo $(* t_{i,1} r_{i,1}) (divd r_{i,2} h_{i,2}) t_{i,1} (+ r_{i,1} h_{i,2}) t_{i,3} -9 8939e-005 -0.00085842 0.16331 0 0032967 -0.1872 -0.15313)$
 (modelo $(divd d_{i,2} h_{i,2}) t_{i,1} t_{i,2} t_{i,3} d_{i,3} -0.12032 0.37987 -0.030729 -0.35936 0.008951 0 12047)$
 (modelo $(+ v_{i,1} d_{i,2}) d_{i,3} (- t_{i,3} t_{i,1}) (cos r_{i,2}) (- r_{i,1} r_{i,3}) -1 5593e-005 -0.0020714 -0.26119 -0.14911 0.0035649 0 085004)$
 (modelo $t_{i,2} t_{i,3} t_{i,1} d_{i,3} t_{i,3} -0.36826 0 0 35842 -0.0013471 0 0 1251)$
 (modelo $(tan r_{i,1}) t_{i,1} (sin v_{i,2}) t_{i,3} d_{i,3} 0.041945 0 34911 -0.048097 -0.36018 -0.001533 0.11753)$
 (modelo $t_{i,3} (* t_{i,1} h_{i,2}) (* h_{i,2} v_{i,2}) t_{i,3} t_{i,1} -0.36662 -4.0679e-005 5.3712e-006 0 0 35772 0.086796)$
 (modelo $d_{i,2} v_{i,3} t_{i,3} (exp d_{i,3}) t_{i,1} -0.0040728 0.00057726 -0.36569 1.4058e-012 0.35266 0.068836)$
 (modelo $(* d_{i,2} r_{i,1}) (* d_{i,2} r_{i,1}) t_{i,3} v_{i,3} t_{i,1} 1.0538e-005 0 -0.3644 0.00052732 0.34851 0.090186)$
 (modelo $(sin h_{i,2}) (sin d_{i,1}) t_{i,1} t_{i,3} t_{i,1} -0.00096118 -0.032294 0.36276 -0.37326 0 0 13025)$
 (modelo $(* t_{i,2} r_{i,2}) (* r_{i,1} t_{i,2}) r_{i,2} (- t_{i,1} r_{i,1}) h_{i,2} 0.0010609 -0.0010226 -0.02423 -0.024765 0.0056963 -0.44093)$

Anexo E

Errores de las mejores propuestas de cada corrida, Caso 2.

Corrida	Norma 1	Norma 2	Norma infinito	No Nodos
1	136.8525	10.2489	2.1312	14
2	137.4477	10.0898	2.1571	16
3	133.61	9.8212	2.1225	19
4	145.7093	10.9435	2.1709	11
5	143.0773	10.6328	2.0727	15
6	142.2088	10.5079	2.1317	17
7	153.7613	11.4529	2.082	14
8	136.3577	10.0973	2.1552	16
9	149.8835	11.0694	2.1309	19
10	153.765	11.4529	2.0802	14
11	137.2876	10.0593	2.1395	16
12	147.9405	11.0786	2.0683	16
13	141.1199	10.3439	2.0908	19
14	154.0301	11.4878	2.1101	17
15	143.2405	10.6357	2.0708	17
16	148.2079	11.074	2.0415	17
17	148.1757	11.0717	2.0331	16
18	153.8496	11.4705	2.1044	18
19	141.5022	10.5411	2.1691	13
20	144.6344	10.5461	2.1791	10

Anexo F.

Diferentes normas obtenidas por grupo, en predicción a largo plazo, Caso 2

Corrida	Norma 1			Norma 2			Norma infinito		
	Grupo 1	Grupo 2	Grupo 3	Grupo 1	Grupo 2	Grupo 3	Grupo 1	Grupo 2	Grupo 3
	-15%	-60%	-40%	-15%	-60%	-40%	-15%	-60%	-40%
1	137.9	908.2	1109	10.33	26.04	38.42	2.118	2.884	4.552
2	156.2	986.3	687.4	11.63	28.84	23.98	2.53	4.282	2.878

3	146.4	942.4	757.4	10.88	27.22	26.39	2.481	3.053	3.178
4	145.6	943.9	994.4	10.93	26.9	33.8	2.154	2.885	4.316
5	143.1	956	1081	10.63	27.08	37.58	2.073	2.937	4.674
6	143.8	944.7	1062	10.68	27.07	36.89	2.29	2.865	4.319
7	153.4	1010	921	11.44	28.62	31.58	2.053	2.89	4.102
8	155.8	988.6	687.1	11.62	28.93	24.1	2.501	4.15	2.936
9	153.2	1001	916.5	11.39	28.67	31.39	2.407	3.195	3.702
10	153.4	1010	921.3	11.44	28.62	31.59	2.051	2.89	4.103
11	157.1	990.2	689	11.63	28.86	23.99	2.412	4.369	2.929
12	151	1008	905.1	11.32	28.71	31.25	2.049	2.99	4.125
13	155.5	1033	736.4	11.57	30	25.81	2.309	3.982	3.14
14	153.5	1004	907.6	11.46	28.57	31.12	2.073	3.048	3.995
15	143.2	954.4	1080	10.64	27.04	37.56	2.071	2.928	4.672
16	151.5	1009	903.2	11.33	28.69	31.06	2.035	2.988	4.092
17	151.5	1004	901.5	11.32	28.68	31.12	2.034	3.016	4.155
18	153.4	1011	925.5	11.45	28.61	31.61	2.068	2.972	4.074
19	151.9	930.5	738.9	11.12	27.02	26.31	2.232	3.333	3.036
20	158.3	1067	765.7	11.69	30.84	26.69	2.238	3.977	3.154

Anexo G

Modelos obtenidos con programación genética con función objetivo de la norma infinito y norma 1

- (modelo $t_{r,1} (+ h_{r,1} t_{r,2}) h_{r,3} (- r_{r,2} r_{r,1})$ 0 062114 -0.0498 0 052955 -0 0087323 -0 30782)
- (modelo $t_{r,1} (+ h_{r,2}) t_{r,3} (\sin v_{r,3}) (\cos r_{r,3})$ 0 34655 0 0024286 -0 35456 -0 04756 -0 08396 -0 01382)
- (modelo $(\sin r_{r,1}) (\sin v_{r,1}) (- r_{r,1} r_{r,2}) (- t_{r,1} t_{r,2}) (\sin h)$ 0 095011 -0 060421 0 0061174 0 29878 0 05374 -0 0046951)
- (modelo $(\cos r_{r,3}) h_{r,3} t_{r,3} h_{r,1} -0.20128$ 0 072125 -0 015033 -0 069478 0 14645)
- (modelo $(\cos r_{r,3}) h_{r,2} (- r_{r,2} r_{r,1}) d_{r,3} h_{r,3} -0 22552 -0 071312 -0 010513 0 0060767 0 07459 -0 06551)$
- (modelo $h_{r,2} t_{r,3} (\sin r_{r,1}) (- r_{r,1} r_{r,2}) (+ h_{r,3} t_{r,1}) -0 073695 -0 066103 0 04725 0 0091405 0 077065 -0 30214)$
- (modelo $t_{r,3} (\cos r_{r,3}) (\sin r_{r,3}) h_{r,2} h_{r,3} -0.014467 -0.27336$ 0 057901 -0 099841 0 10447 0 075441)
- (modelo $h_{r,1} h_{r,3} (- t_{r,3} t_{r,1}) (\sin t_{r,1}) (\cos t_{r,2}) -0 006479 0 0096638 -0.33876 0.052461 0 050121 -0 15216)$
- (modelo $(- h_{r,3} t_{r,2}) (- t_{r,1} h_{r,2}) h_{r,3} (+ d_{r,1} h_{r,2}) (\sin h_{r,2})$ 0 11545 0 11013 -0 013507 0 012927 -0 037076 -0 21173)
- (modelo $t_{r,3} (\cos r_{r,3}) (\sin r_{r,3}) h_{r,2} h_{r,3} -0.01447 -0 27312 0 058748 0 1046 -0 099976 0 075441)$
- (modelo $t_{r,1} (\sin h) (\cos r_{r,3}) t_{r,3} (+ t_{r,1} t_{r,2})$ 1 3452 0 085298 -0 10176 0 63817 -0 99876 0 22254)
- (modelo $(\tan d_{r,1}) h_{r,2} (\cos r_{r,3}) t_{r,1} (- t_{r,2} h_{r,3})$ 0 0019577 -0 098942 -0 20411 0 10995 -0 10703 -0 34476)
- (modelo $(- h_{r,2} t_{r,2}) (+ h_{r,3} t_{r,2}) (- t_{r,1} h_{r,2}) (\sin t_{r,1}) t_{r,3}$ 0 31542 0 052592 0 36506 0 045213 -0 1039 -0 126)
- (modelo $(- h_{r,3} h_{r,2}) (\cos r_{r,3}) (\sin r_{r,3}) (\sin d_{r,1}) t_{r,3}$ 0 10284 -0 26509 0 045759 -0 02003 -0 027261 0 46214)
- (modelo $(+ d_{r,2} d_{r,2}) (- r_{r,1} r_{r,2}) h_{r,2} (\cos t_{r,3}) h_{r,3}$ 0 0024444 0 010494 -0 071399 -0 22695 0 074584 -0 056696)
- (modelo $t_{r,2} (+ h_{r,3} t_{r,1}) (\cos r_{r,3}) (+ d_{r,3} t_{r,1}) h_{r,2} -0 11736 0 1072 -0 21449 0 0045985 -0 10077 -0 16506)$
- (modelo $(- t_{r,1} h_{r,2}) h_{r,3} (\cos r_{r,3}) (\cos v_{r,1}) t_{r,2}$ 0 10348 0 10821 -0 21285 0 059129 -0 11148 -0 044465)
- (modelo $(+ d_{r,2} h_{r,2}) (\sin r_{r,2}) (- h_{r,3} h_{r,2}) t_{r,3} (\cos r_{r,3})$ 0 0038188 0 055095 0 10502 -0 01781 -0 26549 0 14237)
- (modelo $(- h_{r,1} h_{r,3}) t_{r,2} t_{r,3} (\cos r_{r,3}) -0 037306 0 33042 -0 35 -0 14893 0 31118)$

(modelo $(-t_{i-1}, t_{i,2})$ $h_{i,3} h_{i,2}$ 0.46627 0.056481 -0.052984 -0.17482)

Anexo H:

Errores de las mejores propuestas de cada corrida, Caso 3

Corrida	Norma 1	Norma 2	Norma Infinito	No Nodos
1	136.3431	10.1416	2.2644	16
2	122.5665	9.0728	2.4066	18
3	136.3161	10.1491	2.2567	18
4	135.879	10.1245	2.2735	15
5	136.6284	10.1509	2.2452	16
6	131.0385	9.8387	2.4318	20
7	135.8371	10.1141	2.2749	15
8	135.6431	10.1005	2.3077	13
9	133.3685	9.9159	2.2276	20
10	135.3815	10.0905	2.3109	16
11	123.1423	9.4224	2.5269	18
12	136.7738	10.1464	2.2529	14
13	132.6235	9.811	2.3494	19
14	136.437	10.1572	2.2663	12
15	135.6453	10.0742	2.2501	14
16	136.3335	10.1519	2.2816	16
17	136.0897	10.1217	2.2888	13
18	135.8579	10.1172	2.2974	16
19	136.0777	10.1503	2.2497	14
20	117.1992	8.8531	2.3992	18

Anexo I:

Diferentes normas obtenidas por grupo, en predicción a largo plazo, Caso 3.

Corrida	Norma 1			Norma 2			Norma Infinito		
	Grupo 1	Grupo 2	Grupo 3	Grupo 1	Grupo 2	Grupo 3	Grupo 1	Grupo 2	Grupo 3
	-15%	-60%	-40%	-15%	-60%	-40%	-15%	-60%	-40%
1	161.4	1007	703.4	12.01	29.54	24.69	2.657	4.348	2.873
2	298.9	3822	4289	29.41	292.2	287.1	7.411	79.41	64.11
3	161.7	1115	758.9	12.01	33.72	27.03	2.692	4.43	3.297
4	161.4	1013	716.2	12.01	29.66	24.96	2.697	4.285	2.829
5	161.3	1004	700.6	12	29.5	24.6	2.682	4.355	2.814
6	138	906.7	659.8	10.58	26.86	23.47	2.516	3	3.219
7	161.3	1286	863.1	11.92	48.09	35.6	2.684	8.804	8.779
8	156.6	1001	702.2	11.73	29.25	24.57	2.673	4.216	2.793
9	147.2	940.1	749.2	11.02	27.29	26.05	2.441	3.147	3.137
10	157.1	993.1	686.8	11.68	29.06	24.08	2.677	4.272	2.781
11	127.2	884	626.2	9.758	25.69	22.24	2.242	2.766	2.933
12	163	1017	704.2	12.08	29.8	24.72	2.694	4.443	2.826
13	142.1	926.6	762.6	10.61	27	26.4	2.354	3.013	3.117
14	161.1	1003	698.2	11.97	29.45	24.52	2.689	4.368	2.82

15	158.8	993.1	697.3	11.76	29.12	24.59	2.71	4.246	2.967
16	159.8	1002	697.6	11.89	29.34	24.46	2.711	4.319	2.773
17	160	1003	697.8	11.86	29.41	24.39	2.737	4.302	2.76
18	159.3	999.7	693.9	11.81	29.27	24.25	2.714	4.291	2.74
19	162.6	1007	701.8	12.07	29.58	24.63	2.696	4.411	2.822
20	299.6	2704	2844	31.84	150.1	163.9	8.781	32.26	32.03

Anexo J

Relación de los parámetros de vigilancia y pesos obtenidos con algoritmos genéticos

Caso 4

Corrida	Parámetro Vigilancia	Parámetro Vigilancia	Peso 1	Peso 2	Peso 3	Peso 4	Peso 5	Peso 6
1	0.32665	0.99857	0.66707	0.073852	0.0073041	0.0046041	0.16147	0.085697
2	0.54286	0.999	0.95444	0.00098417	0.018243	0.00096267	0.024051	0.0013235
3	0.24543	0.99499	0.77495	0.0027648	0.0039798	0.00081978	0.068917	0.14857
4	0.29455	0.96205	0.63055	0.00069662	0.0031176	0.00075138	0.062275	0.30261
5	0.2581	0.99882	0.7197	0.0010242	0.0040048	0.0020713	0.11696	0.15624
6	0.25728	0.99391	0.46142	0.0716	0.1895	0.01993	0.25148	0.0060684
7	0.25387	0.98071	0.72414	0.012776	0.0017602	0.00077273	0.10606	0.15449
8	0.29732	0.98569	0.82214	0.0091383	0.0035096	0.0020697	0.1287	0.032447
9	0.64887	0.99275	0.30206	0.082216	0.15222	0.019746	0.17229	0.27146
10	0.30015	0.99811	0.62484	0.038182	0.0086893	0.00063334	0.058376	0.26928
11	0.35909	0.99764	0.51936	0.052346	0.010292	0.0011142	0.020664	0.39622
12	0.82019	0.99818	0.25913	0.14946	0.21344	0.02883	0.17604	0.1731
13	0.3861	0.99895	0.47447	0.08249	0.041728	0.0059128	0.12374	0.27166
14	0.22954	0.99347	0.75352	0.00089474	0.0016922	0.0010301	0.10007	0.1428
15	0.36909	0.98441	0.67245	0.0086909	0.013047	0.00069493	0.039946	0.26517
16	0.29811	0.999	0.90746	0.00095769	0.0083193	0.001075	0.081261	0.0009269
17	0.56578	0.97714	0.89642	0.02052	0.019829	0.00092568	0.061206	0.0010989
18	0.28947	0.96782	0.68356	0.0014511	0.0049994	0.00080273	0.088247	0.22094
19	0.28204	0.96126	0.73949	0.0016372	0.004673	0.00078797	0.071266	0.18215
20	0.42003	0.9732	0.67116	0.0237	0.0053764	0.00071756	0.082805	0.21624

Anexo K:

Resultados de las normas durante el proceso de prueba y durante el proceso de entrenamiento
No de patrones y tiempos, para el Caso 4.

Corrida	Entrena		Prueba		No de Patrones	tiempo estimación	tiempo A.G.
	Norma 2	Norma inf.	Norma 2	Norma inf.			
1	78.105	7.7	109.67	8.6	1358	0.8285	49.652
2	410.86	19.549	341.78	19.549	347	0.29833	49.345
3	123.33	8.4	105.85	10.2	726	0.46233	53.274
4	173.54	11.4	144.81	10.733	532	0.37817	47.977
5	127.37	8.3	110.95	9.05	960	0.74233	49.422

6	23.485	3.9	114.07	9.8	1733	1.0783	44.331
7	115.23	11.6	98.876	7.5	782	0.72233	42.413
8	136.58	8.775	132.04	10.75	858	0.61433	43.079
9	49.544	8.2	117.42	8.6	1220	0.7845	42.061
10	108.21	12.714	138.63	11.914	982	0.63533	41.96
11	83.049	6.8	114.4	7.7667	863	0.573	124
12	39.852	6.0091	121.02	10.6	1196	0.822	126.43
13	70.1	9.5	129.24	13.9	1392	0.86967	131.09
14	99.864	7.9667	100.2	7.9	741	0.5555	128.05
15	165.32	13.073	140.99	12.273	564	0.43017	122.85
16	147.7	9.2286	118.09	9.1111	661	0.45967	132.7
17	325.99	17.224	295.29	17.224	430	0.33517	120.46
18	146.82	9.5	132.2	9.8	690	0.47333	103.7
19	135.08	12.933	113.84	9.8	681	1.1425	79.06
20	202.11	14.5	175.64	12.925	449	0.93467	37.564

Anexo I.

Relación de los parámetros de vigilancia y pesos obtenidos con algoritmos genéticos
Caso 5.

Corrida	Parámetro Vigilancia	Parámetro Vigilancia	Peso 1	Peso 2	Peso 3	Peso 4	Peso 5	Peso 6
1	0.026375	0.76167	0.075003	0.24172	0.24512	0.011671	0.27404	0.15244
2	0.034843	0.9358	0.10881	0.2064	0.22604	0.034943	0.30323	0.12058
3	0.023324	0.78493	0.13045	0.13013	0.20477	0.06995	0.25614	0.20855
4	0.038865	0.4726	0.21009	0.18639	0.23982	0.15979	0.14185	0.062064
5	0.030264	0.11522	0.1376	0.2887	0.28443	0.087917	0.189	0.012351
6	0.027972	0.64665	0.15548	0.12493	0.14617	0.10278	0.23015	0.24048
7	0.030647	0.091007	0.21298	0.13074	0.19173	0.10062	0.10309	0.26085
8	0.050354	0.78728	0.12135	0.13497	0.22044	0.12094	0.17568	0.22661
9	0.021084	0.30724	0.059898	0.21544	0.20157	0.083313	0.19222	0.24757
10	0.046359	0.48445	0.073091	0.085048	0.26494	0.0062252	0.32768	0.24301
11	0.027772	0.72283	0.25205	0.095664	0.25795	0.11366	0.069502	0.21117
12	0.027447	0.85792	0.21399	0.071875	0.18614	0.10767	0.20118	0.21915
13	0.030472	0.48393	0.12175	0.20594	0.32692	0.03805	0.062973	0.24436
14	0.032221	0.19523	0.032017	0.23354	0.35029	0.040934	0.076632	0.26658
15	0.018499	0.67264	0.1839	0.17707	0.15969	0.0088602	0.12604	0.34443
16	0.045398	0.69364	0.074916	0.012381	0.38735	0.066281	0.24413	0.21494
17	0.037682	0.46685	0.12803	0.25395	0.22214	0.12862	0.088067	0.17919
18	0.064633	0.54215	0.035103	0.18864	0.15739	0.1092	0.18961	0.32006
19	0.041229	0.2676	0.1157	0.031735	0.24722	0.00085375	0.3837	0.22079
20	0.032274	0.50043	0.0010779	0.19523	0.2968	0.091945	0.16482	0.25012

Anexo M:

Resultados de las normas durante el proceso de prueba y durante el proceso de entrenamiento
No de patrones y tiempos, para el Caso 5.

Corrida	Entrena Norma 2	Norma infinito	Prueba Norma 2	Norma infinito	No de Patrones	tiempo estimación	tiempo para A.G.
1	21.83	2.425	21.466	2.6	611	0.39817	30.669
2	23.413	3.6	22.603	3.3833	450	0.31583	29.912
3	23.45	3.4	23.535	3.5	971	0.56117	32.388
4	26.951	4.4	24.529	3.7	567	0.36433	30.202
5	24.299	4	24.78	3.5	803	0.47967	30.758
6	25.706	3.6	24.679	3.15	773	0.466	31.627
7	26.336	4	22.636	2.8	663	0.41117	30.539
8	23.97	3.8	21.063	2.72	273	0.23517	29.363
9	17.802	3.4	23.17	3.5	1078	0.61417	32.813
10	20.748	2.74	18.336	2.4375	172	0.19683	29.419
11	30.035	4	25.436	3.1	769	0.46233	99.963
12	28.124	2.8	23.521	2.7	792	0.47333	99.939
13	22.812	4	21.791	3.7	497	0.3305	93.722
14	18.48	1.9	20.138	2.6	419	0.303	93.882
15	28.106	3.6	25.433	3.5	968	0.55833	102.36
16	21.219	2.3286	18.685	2.8	232	0.226	91.375
17	23.671	4	22.469	2.8	523	0.45867	92.588
18	21.266	2.5333	19.254	2.45	160	0.24267	90.477
19	22.097	2.6286	19.001	3	207	0.26917	91.196
20	17.741	2.2816	21.359	2.6	575	0.3955	94.639

Anexo N:

Relación de los parámetros de vigilancia y pesos obtenidos con algoritmos genéticos
Caso 6.

Corrida	Parámetro Vigilancia	Parámetro Vigilancia	Peso 1	Peso 2	Peso 3	Peso 4	Peso 5	Peso 6
1	0.79263	0.99524	0.23848	0.24126	0.2436	0.02384	0.082615	0.17021
2	0.43009	0.99558	0.12687	0.26091	0.16445	0.019776	0.24193	0.18606
3	0.41252	0.98397	0.18616	0.30255	0.19139	0.023301	0.081834	0.21476
4	0.46169	0.995	0.23904	0.32197	0.23238	0.025944	0.080755	0.099909
5	0.40089	0.99896	0.20446	0.30109	0.19186	0.023484	0.074138	0.20496
6	0.3239	0.98985	0.14118	0.24387	0.14557	0.018227	0.273	0.17816
7	0.47503	0.99499	0.064406	0.29758	0.30371	0.033289	0.16392	0.13709
8	0.37728	0.99453	0.197	0.28392	0.18439	0.020831	0.25776	0.056105
9	0.94091	0.9984	0.19404	0.17411	0.19939	0.021929	0.20546	0.20507
10	0.47986	0.99823	0.27383	0.34885	0.21116	0.02536	0.075196	0.065604
11	0.99377	0.9983	0.32881	0.16882	0.02698	0.0010382	0.17885	0.29551
12	0.78824	0.99846	0.16701	0.16655	0.018607	0.00044888	0.29212	0.35527
13	0.34571	0.98843	0.21508	0.19076	0.21032	0.022739	0.18197	0.17912
14	0.72528	0.99393	0.5266	0.0039839	0.0054368	0.0091533	0.032151	0.42268

15	0.99631	0.99832	0.3084	0.2223	0.024657	0.00058213	0.28237	0.1617
16	0.81623	0.99865	0.23307	0.15406	0.01464	0.0016801	0.28529	0.31126
17	0.88629	0.96639	0.11609	0.16662	0.018012	0.0025295	0.30022	0.39653
18	0.94705	0.99899	0.35448	0.015214	0.08598	0.00060246	0.3606	0.18313
19	0.45597	0.99688	0.34123	0.0013564	0.0030755	0.00062219	0.26712	0.38659
20	0.4715	0.99883	0.26598	0.34067	0.21813	0.024461	0.10924	0.041509

Anexo O

Resultados de las normas durante el proceso de prueba y durante el proceso de entrenamiento
No de patrones y tiempos, para el Caso 6.

Corrida	Entrena Norma 2	Norma infinito	Prueba Norma 2	Norma infinito	No de Patrones	tiempo estimación	tiempo para A.G.
1	22.513	2.4	24.428	3.2	1136	1.2167	44.166
2	18.511	2.4	24.62	3.2	1519	1.6925	50.105
3	20.847	2.4	25.374	3.2	1567	1.7247	55.886
4	22.584	2.4	25.529	3.2	1524	1.6788	50.002
5	21.987	2.4	25.202	3.2	1580	1.7868	60.583
6	19.885	2.4	24.787	3.2	1661	1.8337	49.017
7	12.436	1.5667	25.749	3.5	1467	1.6705	45.726
8	21.675	2.4	24.549	3.2	1604	1.829	52.303
9	20.928	2.4	22.557	3.2	951	1.1213	44.394
10	23.955	2.4	25.939	3.2	1516	1.678	43.919
11	27.563	4	21.482	2.8	274	0.49067	111.09
12	23.95	2.95	20.275	2.8	316	0.52183	109.93
13	23.414	3.1	25.461	3.2	1598	1.7302	129.94
14	47.047	3.5333	37.787	3.2	191	0.41933	103.58
15	26.174	4	21.446	2.8	315	0.52	111.09
16	26.904	4	22.411	2.8	270	0.5155	105.79
17	22.102	3.0167	19.12	2.8	235	0.44583	107.59
18	35.395	3.7889	28.437	4.7	340	0.531	110.04
19	31.581	2.4958	25.603	2.4958	249	0.47417	88.52
20	23.738	2.4	25.918	3.2	1518	1.677	85.172

Anexo P

Normas por corrida durante cada fase Caso 7.

Corrida	Creacion		Entrena		Prueba		Tiempo A. Genético
	Norma 2	Norma infinito	Norma 2	Norma infinito	Norma 2	Norma infinito	
1	27.89	2.654	71.37	7.977	29.78	3.543	15.5
2	49.49	4.615	40.07	5.082	40.74	6.352	15.05
3	42.89	3.754	129.3	27.47	36.28	3.732	14.5
4	54.08	3.274	57.32	8.337	44.31	5.841	14.45
5	31.78	2.46	42.8	3.363	24.42	3.776	15.07
6	28.53	3.299	23.15	3.299	1256	1073	14.78

7	42.64	4.152	42.55	6.147	38.57	3.667	14.6
8	54.57	3.162	120.5	24.25	44.22	4.991	14.43
9	29.37	2.566	59.07	50.84	22.96	4.286	14.44
10	26.39	2.786	160	47.18	22.14	2.731	14.44
11	32.08	3.843	95.85	24.13	32.11	4.51	13.5
12	60.57	3.498	60.35	5.59	50.49	5.982	13.2
13	70.77	5.959	126.9	43.28	58.46	7.874	12.92
14	97.56	9.719	2455	824.4	82.93	18.82	12.93
15	65.67	4.277	891.9	240	55.78	7.557	12.91
16	40.74	3.298	257.9	85.61	33.22	5.496	12.86
17	88.53	5.338	235.3	47.99	72.33	11.47	13.25
18	64.02	9.438	91.96	9.607	44.73	4.956	13.34
19	45.95	2.991	210.4	105.4	39.96	3.935	13.86
20	45.35	3.242	35.33	3.003	36.29	3.485	13.06